

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
INFORMÁTICA
INGENIERÍA INFORMÁTICA

**APLICACIÓN WEB PARA EL APOYO A LA DOCENCIA DE
BASES DE DATOS ORACLE**

**WEB APPLICATION FOR THE SUPPORT TO THE ORACLE
DATABASE TEACHING**

Realizado por
Antonio Ángel Gamarro Orozco
Tutorizado por
Enrique Soler Castillo
Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, Septiembre 2018

Fecha defensa:
El secretario del Tribunal

Resumen

El objetivo de este trabajo de fin de grado es el desarrollo de una aplicación web para el apoyo a la docencia de bases de datos Oracle, la cual puede ser útil para asignaturas de varias titulaciones de la Escuela Técnica Superior de Ingeniería Informática.

La aplicación tendrá dos partes, una para los alumnos y otra para los docentes. Se centrará en los principales tipos de pruebas de esta materia, los cuales son DML (lenguaje de manipulación de datos) y DDL (lenguaje de definición de datos).

Desde la parte de alumnos, se podrá solicitar el usuario vinculado a la matrícula en una asignatura y, una vez identificado, realizar pruebas y ver las correcciones cuando estén disponibles.

Desde la parte de docentes, se podrán gestionar las asignaturas, los alumnos, las matrículas, las pruebas y las cuentas de Oracle. Las pruebas se podrán asignar a las distintas asignaturas para ciertos periodos de tiempo y hacer correcciones tanto automáticas como manuales, así como bloquear y desbloquear las cuentas de Oracle vinculadas a los alumnos de una prueba asignada.

Además, se podrán generar informes tanto predefinidos (generalmente de calificaciones) como personalizados (mediante consultas SQL), con posibilidad de exportarlos en varios formatos.

Palabras clave

Bases de datos, Oracle, docencia, DML, DDL, aplicación web, evaluación de alumnos

Abstract

The objective of this end-of-degree project is the development of a web application to support the teaching of Oracle databases, which can be useful for subjects of various degrees of our University.

The application will have two parts, one for the students and another for the teachers. It will focus on the main types of tests of this subject, which are DML (data manipulation language) and DDL (data definition language).

From the student's side, the user linked to the enrolment of a student to a subject can be requested and, once identified, the student can perform tests and see the corrections when they are available.

From the teacher's side, subjects, students, enrolments, tests and Oracle accounts can be managed. Tests can be assigned to the different subjects for certain periods of time and perform both automatic and manual corrections, as well as locking and unlocking the Oracle accounts linked to the students of an assigned test.

In addition, reports can be generated both predefined (about qualifications) and customized (through SQL queries), with the possibility of exporting them in different formats.

Keywords

Databases, Oracle, teaching, DML, DDL, web application, student evaluation

Índice

1	Introducción.....	9
1.1	Marco contextual	9
1.2	Objetivos y motivación	10
1.3	Tecnologías y herramientas utilizadas	10
2	Análisis de requisitos.....	15
2.1	Actores	15
2.2	Requisitos funcionales	15
2.3	Requisitos no funcionales	19
3	Diseño de la aplicación	21
3.1	Diagramas de casos de uso.....	21
3.1.1	Panel de docente.....	22
3.1.1.1	Gestión de asignaturas	23
3.1.1.2	Gestión de alumnos	25
3.1.1.3	Gestión de pruebas.....	27
3.1.1.4	Gestión de notas externas	30
3.1.1.5	Gestión de cuentas	32
3.1.1.6	Informes	33
3.1.1.7	Configuración	34
3.1.2	Panel de alumno.....	35
3.2	Base de datos	36
3.2.1	Esquema	37
3.2.2	Elementos PL/SQL	40
4	Implementación.....	43
4.1	Metodología de desarrollo.....	44
4.2	Estructura de la aplicación	46

4.3	Núcleo de la aplicación	50
4.4	Seguridad.....	54
4.5	Aspecto visual e interacción con el usuario.....	56
4.6	Configuración de la base de datos.....	59
4.7	Infraestructura	61
4.8	Pruebas.....	62
4.8.1	Pruebas funcionales	62
4.8.2	Pruebas de carga	69
5	Conclusiones y trabajo futuro.....	75
6	Referencias	77
7	Anexos	79
7.1	Manual de usuario.....	79
7.1.1	Manual para usuarios con rol Docente	79
7.1.2	Manual para usuarios con rol Alumno	87
7.2	Manual de instalación.....	90
7.3	Manual para añadir nuevas funcionalidades	92
7.4	Manual de configuración del servidor.....	96

1 Introducción

Este documento se corresponde con la memoria del Trabajo de Fin de Grado titulado “Aplicación de apoyo a la docencia de base de datos Oracle”, en el cual se desarrollará una descripción técnica del proyecto y se hará un recorrido de las principales fases llevadas a cabo para su desarrollo.

En este primer capítulo se contextualizará el proyecto tanto de forma general, fijando los objetivos a conseguir y los hechos que motivan su realización, como tecnológicamente, describiendo las tecnologías, métodos y herramientas utilizadas para desarrollarlo e implementarlo.

1.1 Marco contextual

Las bases de datos es una de las materias principales en el ámbito de la Ingeniería Informática, por lo que en esta Universidad existen varias asignaturas dedicadas a ellas, aunque en este trabajo la protagonista es la asignatura Bases de Datos.

En esta asignatura se imparten los principales conceptos de esta materia, entre los que se encuentran el uso de SQL, que es el lenguaje utilizado para el control e interacción con los SGBD (sistemas de gestión de bases de datos). Las operaciones de este lenguaje se pueden dividir en dos tipos, DML (lenguaje de manipulación de datos) y DDL (lenguaje de definición de datos).

Para la evaluación de los conocimientos en esta parte de la asignatura se realizan pruebas prácticas en las que el alumno tiene que especificar las instrucciones necesarias para obtener ciertos resultados en el caso de DML, o utilizar convenientemente las operaciones para generar determinados resultados en una cuenta específica de la base de datos en caso de DDL.

Hasta ahora estas pruebas se realizaban en ordenador, pero con una aplicación muy antigua y con problemas de rendimiento, limitándose a recoger las respuestas de los alumnos a través de un formulario. Para ver las soluciones de los alumnos y realizar el proceso de corrección, el docente tenía que acceder a los datos directamente desde la base de datos.

Es aquí donde surge la necesidad de una aplicación que gestione el proceso de evaluación de los alumnos para este tipo de pruebas, el cual es bastante tedioso de la forma en que se realiza actualmente.

1.2 Objetivos y motivación

El objetivo principal del proyecto es desarrollar una aplicación web para el apoyo a la docencia de base de datos Oracle, automatizando la evaluación de las pruebas prácticas de las asignaturas asociadas a dicha materia.

Con esto buscamos mejorar la calidad de la evaluación de la parte práctica de dichas asignaturas, beneficiando tanto al docente como al alumno.

También se busca facilitar la labor del docente, ya que tanto la preparación y control de los usuarios de base la base de datos necesarios, como la corrección manual de este tipo de pruebas prácticas, resulta tediosa y repetitiva, pudiendo dar lugar a errores. Con la aplicación podrá gestionar de forma organizada las pruebas y entregas de los distintos grupos, simplificando mucho el proceso y pudiendo obtener las calificaciones medias de cada alumno directamente.

De cara al alumno se resolverán los problemas de lentitud y operatividad que ocurren en los exámenes con la herramienta usada hasta ahora, además de tener un portal donde ver su histórico de calificaciones y revisar la corrección de las pruebas, que se podrá hacer antes que si el proceso se realiza manualmente.

La motivación a realizar este trabajo es el estudio de los principales elementos de una aplicación web y la interacción entre ellos de forma general, además de profundizar en las tecnologías escogidas para su desarrollo.

1.3 Tecnologías y herramientas utilizadas

En este punto se identifican las tecnologías en las que se basa la aplicación y las herramientas utilizadas para el desarrollo del proyecto.

Oracle Database

Sistema de gestión de bases de datos relacionales de licencia privativa que destaca en bases de datos empresariales por su alta seguridad y rendimiento.

Para la fase de desarrollo del proyecto se ha utilizado la versión XE (Express Edition), que es gratuita y totalmente funcional a cambio de una serie de limitaciones de rendimiento que no nos afectan en esta fase.

PHP

Lenguaje de programación interpretado de lado del servidor, orientado a objetos, multiplataforma y de código abierto, que se utiliza mayormente para el desarrollo de aplicaciones web. Su baja curva de aprendizaje y el poder embeberlo en código HTML son algunas de las características por las que es tan popular.

En la aplicación se han utilizado varias librerías de terceros escritas en este lenguaje, todas con licencias de libre uso, para realizar algunas tareas concretas:

- GUMP [1]: librería para validar datos de entrada que cuenta con una gran cantidad de reglas y filtros de validación de forma predefinida.
- PhpSpreadsheet [2]: permite leer y escribir hojas de cálculo en los principales formatos utilizados, como CSV, XLS, XLSX y ODS.
- Php-csrf [3]: sencilla librería para manejar tokens de protección contra ataques de tipo CSRF (falsificación de petición en sitios cruzados).

HTML, CSS y Javascript

Trío de tecnologías utilizadas para presentar el contenido de la aplicación y permitir la interacción con el usuario. HTML es un lenguaje de marcado y define la estructura de los datos devueltos al usuario, mientras que CSS es un lenguaje de estilos que define cómo se verán los datos y Javascript es el lenguaje de programación que determinará la lógica en el lado del cliente.

Para facilitar la generación de vistas y mantener la compatibilidad entre los navegadores, usamos el framework **Bootstrap** [4] que combina estos tres lenguajes.

Además, se han usado varias librerías de terceros escritas en Javascript para algunas funcionalidades concretas de lado del cliente:

- jQuery: biblioteca que simplifica la escritura de código Javascript de lado del cliente, permitiendo hacer lo mismo escribiendo menos código y sin necesidad de discriminar entre los distintos navegadores.

- Datable: enriquece las tablas HTML añadiéndoles funcionalidades y controles para la interacción con el usuario.
- nicEdit: simple y rápido editor WYSIWYG para sitios web, que permite dar estilos a los textos como en un procesador de textos para posteriormente guardar el código HTML generado.
- Bootstrap-toggle, bootstrap-select, bootstrap-datetimepicker y bootstrap-editable: plugins para el framework Bootstrap que añaden tipos de controles adicionales para mejorar la experiencia del usuario.

Apache

Servidor web de código abierto y multiplataforma. Cuenta con una arquitectura modular que permite extenderlo y utilizar las funciones requeridas, como en este caso *mod_ssl* para mantener conexiones seguras por HTTPS y *mod_php* para interpretar y ejecutar código PHP. Es el servidor HTTP más utilizado en internet, por lo que tiene una gran comunidad que le brinda soporte.

CentOS

Sistema operativo de código abierto basado en la distribución *Red Hat Enterprise Linux*. Se ha elegido por tener como principales características estabilidad y alta seguridad, ofreciendo prácticamente un producto de nivel empresarial pero gratuito.

Netbeans

Entorno de desarrollo integrado (IDE) libre y multiplataforma que permite programar distintos tipos de aplicaciones en multitud de lenguajes. Entre ellos está el desarrollo de aplicaciones web con PHP, soportando todas sus versiones y ofreciendo un potente debugger integrado que facilita el desarrollo.

Ofrece multitud de características atractivas como coloreado, autocompletado y comprobación de código, sugerencias, integración con control de versiones, gestión de bases de datos, herramientas de refactorización... las cuales simplifican el desarrollo de pequeños y grandes proyectos.

Oracle DataModeler

Herramienta gratuita para el modelado de bases de datos ofrecida por Oracle, la cual utilizamos para realizar el diagrama Entidad-Relación a partir del cual

podremos generar automáticamente el modelo Relacional y el código para crear el esquema de la base de datos.

Oracle SQL Developer

Entorno de desarrollo integrado (IDE) gratuito gestionar y administrar cómodamente bases de datos Oracle locales y remotas, permitiendo la ejecución de consultas y scripts SQL y PL/SQL, acceso a la consola DBA, generación de informes, y muchas más opciones.

Apache JMeter

Aplicación de código abierto para realizar pruebas de carga sobre servicios, especialmente aplicaciones web. Permite preparar escenarios de pruebas totalmente personalizados, tanto a nivel de flujo como a nivel de carga, y obtener medidas y analíticas de rendimiento en base a su ejecución.

Bitvise SSH Client

Cliente SSH gratuito para entornos Windows, que nos permite conectarnos de forma segura a equipos remotos, en este caso el servidor de la aplicación. Nos permite gestionar perfiles de conexión, acceso a la consola de la máquina remota y una interfaz avanzada para transferir archivos a través de SFTP.

VMware Workstation

Hipervisor de tipo 2 (o *hosted*, que se ejecuta sobre otro sistema operativo) de licencia comercial. Utilizado para crear la máquina virtual que hará de servidor desde el equipo de desarrollo, ejecutarla durante la fase de desarrollo y después exportarla y desplegarla en un hipervisor de tipo 1 donde se ejecutará la aplicación en su entorno de producción.

Se ha usado a pesar de su licencia comercial gracias a que la Universidad de Málaga proporciona licencias para los estudiantes.

Draw.io

Aplicación gratuita para la realización de diagramas de todo tipo, con la que se han dibujado la mayoría de los diagramas que aparecen en este documento, que en su mayoría son de tipo UML.

2 Análisis de requisitos

Una vez situado el proyecto en su contexto y tener una idea general de lo que se quiere, la primera fase del proyecto es la obtención de los requisitos, que determinarán concretamente la funcionalidad y el alcance de la aplicación, así como otras características técnicas no menos importantes.

2.1 Actores

Se distinguen dos tipos de usuarios que harán uso de la aplicación:

- **Docente.** O profesor, es el usuario más privilegiado y encargado de gestionar todas las entidades de la aplicación y realizar el proceso de evaluación de los alumnos.
- **Alumno.** Aunque con menos funcionalidades disponibles, es el protagonista de la aplicación puesto que el objetivo de esta es evaluar sus conocimientos. Tendrán acceso a la aplicación sólo para realizar las pruebas asignadas a las asignaturas en las que está matriculado y ver sus calificaciones una vez corregidas.

2.2 Requisitos funcionales

RF01 – Autenticación distinguida para los distintos tipos de usuario.

RF02 – La aplicación tendrá dos paneles distinguidos, uno para el docente y otro para el alumno.

Docente

RF03 – Se deben poder gestionar las asignaturas en las que se utilice la aplicación con datos identificativos como nombre, curso, grupo y titulación.

RF04 – Se gestionarán los datos de los alumnos identificados por número de expediente y DNI/NIE.

RF04.1 – Un alumno podrá matricularse en una o varias asignaturas.

RF04.2 – Se permitirá importar un conjunto de alumnos mediante un fichero CSV o Excel y matricularlos todos a una misma asignatura.

RF04.3 – Al matricular a un alumno en una asignatura se le generará automáticamente su usuario para acceder a la aplicación, válido sólo para esa asignatura.

RF04.4 – Se podrá ver el usuario de cada alumno y cambiar su contraseña.

RF05 – Se deben poder gestionar las pruebas que serán realizadas a través de la aplicación, con un número de preguntas no predefinido.

RF05.1 – De cada pregunta se almacenará el enunciado, la solución y la puntuación.

RF05.2 – Habrá dos tipos de pruebas, DML y DDL.

RF06 – Una prueba se podrá asignar a varias asignaturas, pero sólo una vez a cada una, a una de sus convocatorias.

RF06.1 – Para cada asignación se especificará la fecha de inicio y fin de la prueba y una palabra de paso opcional.

RF06.2 – La asignación se podrá hacer visible o ocultarla de cara a los alumnos.

RF06.3 – Una vez asignada una prueba, no se podrá modificar ni su tipo ni su número de preguntas, pudiendo únicamente corregir erratas.

RF07 – Se podrán gestionar usuarios de la propia base de datos que serán utilizados por los alumnos para realizar las pruebas.

RF07.1 – Habrá dos tipos de usuarios de base de datos, al igual que en las pruebas.

RF07.2 – Para cada alumno se creará un usuario de tipo DML cuando se matricula en una asignatura, el cuál usará para realizar todas las pruebas de tipo DML en esa asignatura y para realizar las prácticas. El nombre de usuario y clave será el mismo que el de acceso a la aplicación.

RF07.3 – Los usuarios de tipo DDL serán creados al asignar una prueba a una asignatura, uno para cada alumno y prueba. Se utilizan sólo para una prueba en concreto. El nombre de usuario se generará en base a un prefijo establecido para la prueba y el número de matrícula, y la clave aleatoriamente.

RF07.4 – Los usuarios podrán bloquearse y desbloquearse ya sea individualmente o por asignatura en caso de los de tipo DML y por asignatura y prueba en caso de los de tipo DDL.

RF07.5 – Se podrá ver a quien pertenece cada usuario y, en el caso de los usuarios de tipo DDL, ver la contraseña.

RF07.6 – Se podrán listar los usuarios filtrados por asignatura en caso de los de tipo DML y por asignatura y prueba en caso de los de DDL.

RF08 – Las entregas de los alumnos podrán ser corregidas mediante la aplicación.

RF08.1 – Se podrán corregir las entregas de los alumnos de una asignatura para una prueba automáticamente mediante procedimientos almacenados de la base de datos.

RF08.2 – Las entregas podrán ser revisadas y calificadas de forma manual, prevaleciendo esta nota sobre la generada automáticamente.

RF08.3 – Se podrá ver la IP desde donde el alumno accedió a la prueba aunque no la haya entregado.

RF09 – Para cada asignatura se podrán gestionar las notas obtenidas por sus alumnos matriculados en pruebas que han realizado utilizando medios externos a la aplicación.

RF09.1 – Se almacenará un registro de las pruebas externas vinculadas a cada asignatura.

RF09.2 – Para cada prueba externa realizada por los alumnos de una asignatura, se podrán introducir las calificaciones individualmente o mediante un fichero con el listado de todas las notas para esa prueba.

RF10 – Será necesario poder calcular las notas finales de cada alumno para cada una de las convocatorias, basándose tanto en las notas de las pruebas realizadas en la aplicación como en las de las pruebas externas.

RF11 – Será posible generar informes con la posibilidad de exportarlos.

RF11.1 – Se podrán generar informes de calificaciones por prueba asignada a una asignatura y por convocatoria de una asignatura, mostrando la calificación para cada alumno matriculado.

RF11.2 – Se podrán generar informes personalizados mediante una consulta SQL de tipo SELECT sobre el esquema de la base de datos de la aplicación.

RF11.3 – Los informes podrán exportarse a algún formato imprimible.

RF12 – El docente podrá añadir enlaces externos al menú que se muestra al alumno cuando está autenticado en la aplicación.

Alumnos

RF13 – Los alumnos podrán solicitar su usuario para acceder a la aplicación una vez que el profesor los matricula en una asignatura, permitiéndole elegir una contraseña.

RF14 – Un alumno podrá recordar su usuario siempre y cuando lo haya solicitado anteriormente.

RF15 – El alumno podrá realizar las pruebas asignadas a su asignatura cuando estén visibles y la fecha y hora actual esté entre las fechas de inicio y fin de la asignación.

RF15.1 – El alumno tendrá que introducir una palabra de paso en caso de que se haya establecido para la asignación.

RF15.2 – El alumno podrá introducir, opcionalmente, el identificador del equipo donde va a realizar la prueba.

RF15.3 – El alumno podrá acceder a realizar la prueba varias veces mientras no efectúe la entrega.

RF16 – El alumno podrá ver las calificaciones obtenidas en las pruebas que haya realizado cuando estén corregidas y siempre y cuando la asignación esté visible.

RF16.1 – El alumno podrá revisar la calificación de cada pregunta junto con el enunciado, la solución y su respuesta.

2.3 Requisitos no funcionales

RNF01 – Se debe usar Oracle como base de datos.

RNF02 - El sistema tendrá que manejar y controlar los accesos concurrentes a la base de datos.

RNF03 – La aplicación debe ser segura, evitando acciones indeseables y acceso a la información sin la autenticación correspondiente.

RNF04 – Las contraseñas de los usuarios de los alumnos deben almacenarse cifradas en la base de datos.

RNF05 – Para la autenticación de los docentes se utilizarán usuarios de la propia base de datos con un rol específico, en vez de almacenarlos en el esquema de la base de datos de la aplicación como para los alumnos.

RNF06 – Debe ser sencillo modificar la aplicación y añadir nuevas funcionalidades.

RNF07 – La aplicación debe soportar la carga necesaria para que los alumnos de una asignatura realicen una prueba de forma concurrente.

RNF08 – La aplicación tiene que ser accesible tanto desde dentro como desde fuera de la universidad.

RNF09 – El sistema tendrá una interfaz clara e intuitiva que facilite su uso tanto al profesorado como a los alumnos.

3 Diseño de la aplicación

En este capítulo se define la funcionalidad y estructura del sistema en base a los requisitos obtenidos en el capítulo anterior y a sus necesidades implícitas.

3.1 Diagramas de casos de uso

A continuación, se definen los diagramas que casos de uso, que describen qué puede hacer cada tipo de usuario o actor de la aplicación.

Los casos de uso de fondo blanco se corresponden con subsistemas o secciones que se detallarán a su vez con otro diagrama de uso, mientras que los de fondo sombreado equivalen a casos de uso finales o acciones atómicas para el usuario.

Debido a la cantidad de funcionalidades de la aplicación, no se detallará cada uno de los casos de uso, aunque si se hará una descripción de cada diagrama que refleje el alcance y las posibilidades de la aplicación.

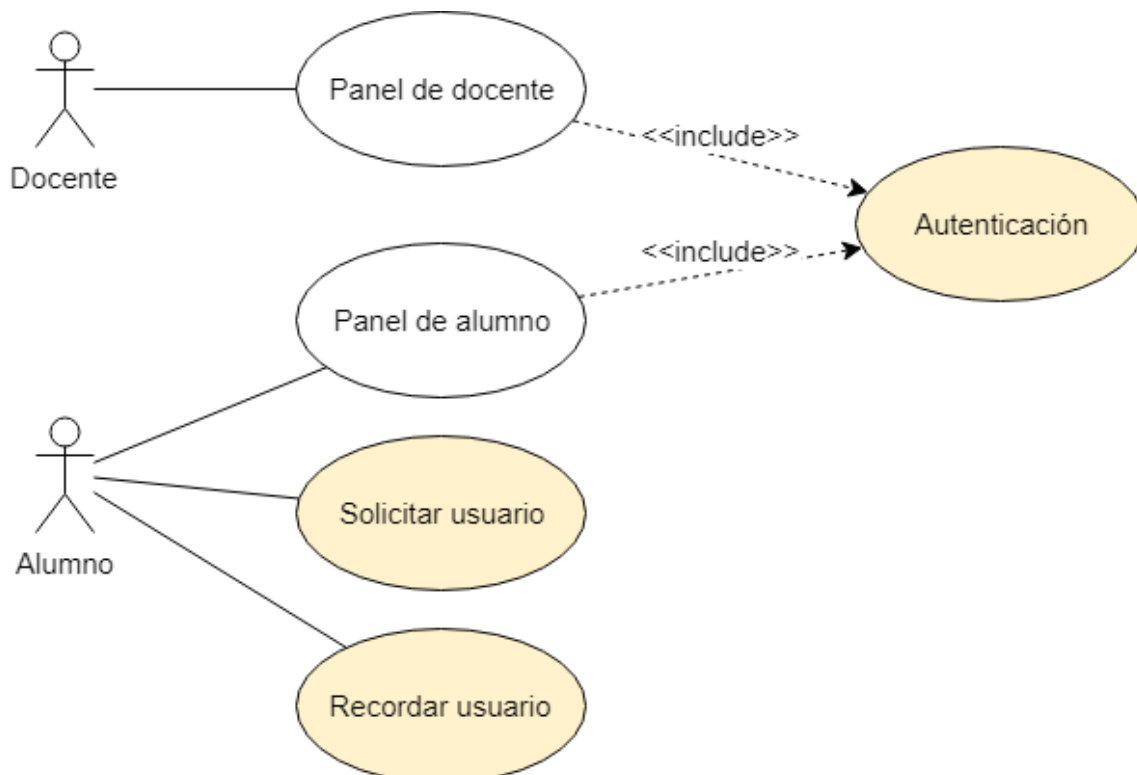


Ilustración 3.1: Diagrama de caso de uso principal

Este primer diagrama representa el inicio de la aplicación, compartido por todos los tipos de actores y permitiendo el acceso al subconjunto de casos de casos de uso que corresponde a cada uno de ellos mediante la autenticación.

Además, en el caso de los alumnos, podrán solicitar su usuario por primera vez o recordarlo si ya lo solicitó anteriormente.

3.1.1 Panel de docente

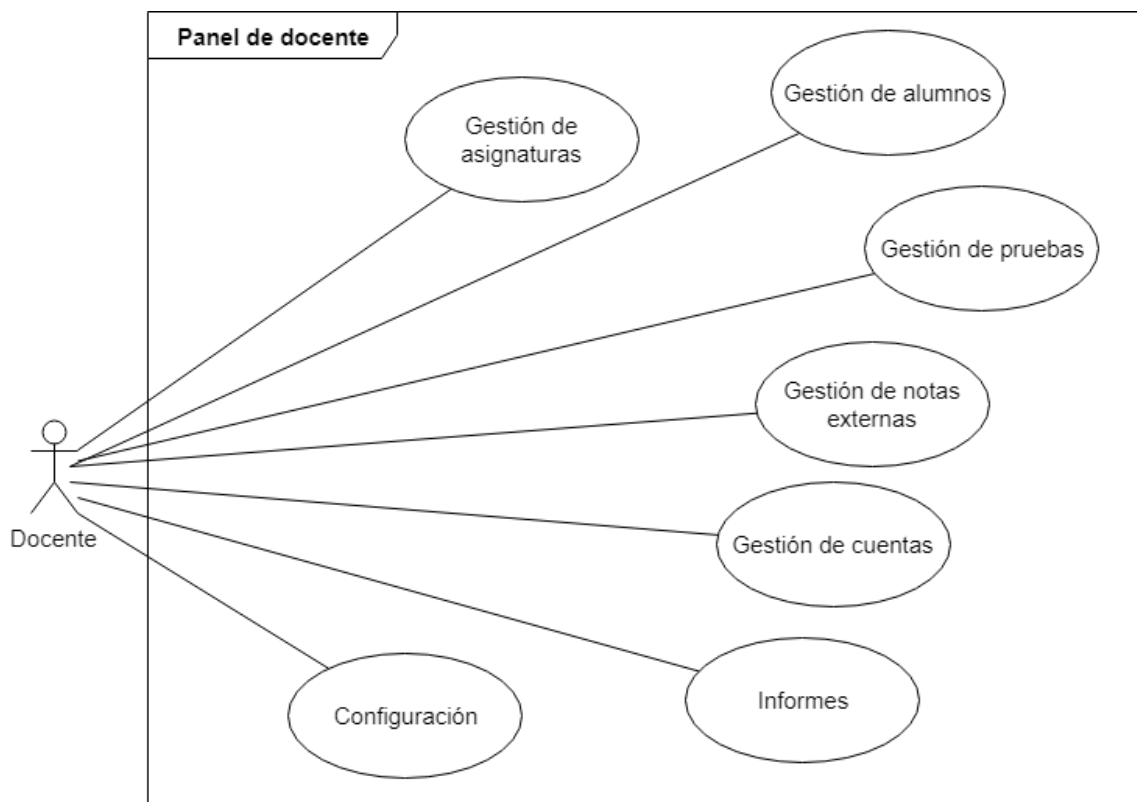


Ilustración 3.2: Diagrama de caso de uso del panel de docente

Una vez el usuario de tipo Docente se autentica en la aplicación, accederá a un panel donde los casos de uso que le corresponden se encuentran agrupados por secciones, de modo que sea más sencillo e intuitivo encontrar una acción concreta de entre todas las que puede realizar.

3.1.1.1 Gestión de asignaturas

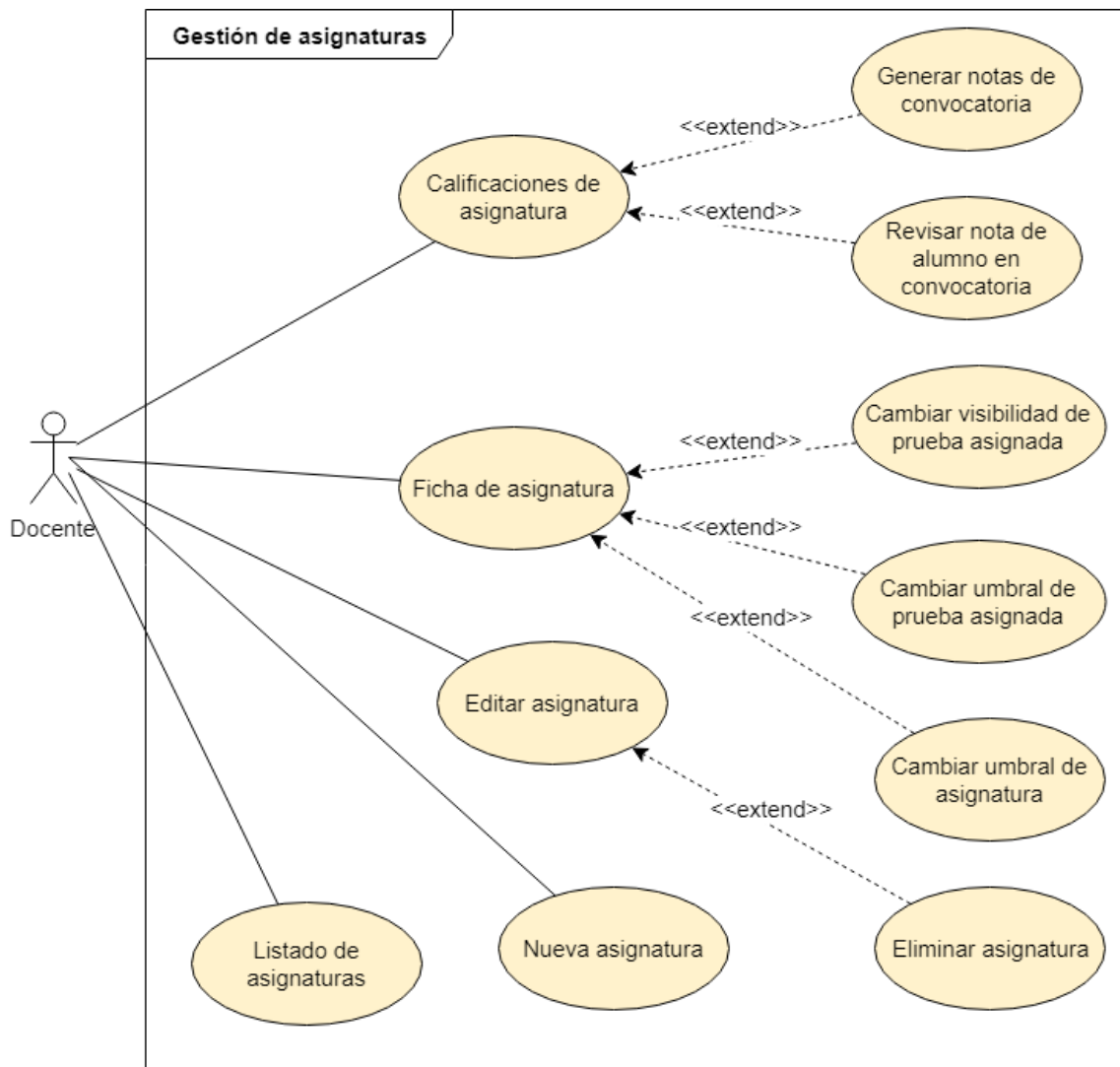


Ilustración 3.3: Diagrama de caso de uso de la gestión de asignaturas

En esta sección se gestionarán las asignaturas, que es la entidad base de la aplicación en torno a la que giran todas las demás.

Listado de asignaturas

Muestra las asignaturas existentes agrupadas por curso académico y titulación y con posibilidad de buscar por una cadena. Desde este listado se accede al resto de casos de usos que están relacionados con una asignatura ya existente:

Editar asignatura

Permite editar los datos de una asignatura o eliminarla.

Ficha de asignatura

Muestra los datos de la asignatura y las pruebas asignadas para cada convocatoria. Además, se permite:

- **Cambiar umbral de asignatura**, que indica el porcentaje de calificación con respecto al valor de cada prueba que el alumno tiene que superar para aprobar la asignatura.
- **Cambiar umbral de prueba asignada**, que se refiere a lo mismo que el anterior pero para una prueba asignada concreta, teniendo este valor prevalencia sobre el de la asignatura.
- **Cambiar visibilidad de prueba asignada**, que permite mostrar u ocultar la prueba para los alumnos de esa asignatura, ya sea para realizarla o para ver las calificaciones (dependiendo de su estado).

Calificaciones de asignatura

Listado de notas finales de la asignatura para cada convocatoria, las cuales se calculan en base a las calificaciones obtenidas en las pruebas asignadas y en las pruebas externas. Cuando las calificaciones para una convocatoria estén generadas, se mostrarán las fechas de generación y última revisión manual.

Para cada alumno, además de su calificación, se mostrarán indicadores en caso de que su nota haya sido modificada manualmente o esté suspenso debido a que no superó el umbral en alguna de las pruebas aun saliendo la media aprobada. Para cada convocatoria se permitirá:

- **Generar notas de convocatoria**, lo cual calcula la calificación final de cada alumno y la almacena. En caso de estar ya generadas, se pueden volver a generar eliminándose las generadas anteriormente.
- **Revisar nota de alumno en convocatoria**, que mostrará las calificaciones parciales de las pruebas para el alumno en esa convocatoria y las calificaciones finales del alumno en el resto de las convocatorias, permitiendo cambiar manualmente la nota del alumno para esa convocatoria y añadir un comentario.

Nueva asignatura

Permite crear una nueva asignatura con todos sus datos.

3.1.1.2 Gestión de alumnos

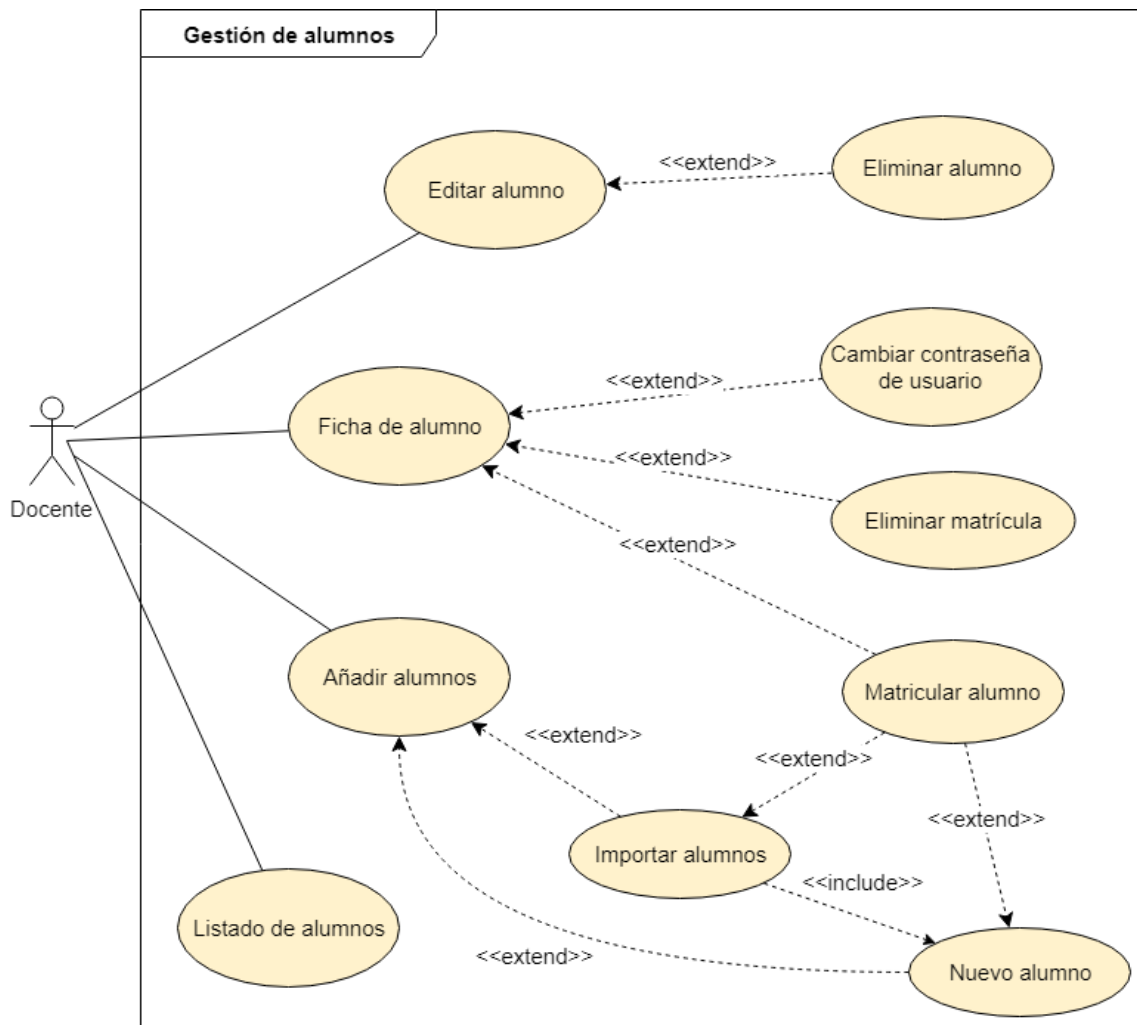


Ilustración 3.4: Diagrama de caso de uso de la gestión de alumnos

Gestión de los alumnos matriculados en una o varias de las asignaturas que utilizan la aplicación para su proceso de evaluación.

Listado de alumnos

Permite obtener un listado de alumnos en base a uno de los filtros predefinidos (todos, sin matricular, matriculados en un curso académico concreto o matriculados en una asignatura concreta). De cada alumno se mostrará su información identificativa (apellidos, nombre, expediente y DNI) y se permitirá ir a su ficha o a la edición de sus datos.

Añadir alumnos

Donde se pueden crear alumnos en la aplicación.

- **Nuevo alumno**, permite crear un alumno de forma individual introduciendo sus datos identificativos en un formulario y, opcionalmente, eligiendo una asignatura a la que matricularlo directamente.
- **Importar alumnos**, permite añadir un grupo de alumnos mediante la subida de un fichero que contenga los datos identificativos de cada uno de ellos. Admite ficheros de tipo CSV y EXCEL con las columnas y el orden de las mismas que se especifican, además de algunas configuraciones opcionales (como desechar la primera fila en caso de tener encabezados y especificar el delimitador en caso de CSV) y la posibilidad de matricularlos directamente en una asignatura.

Ficha de alumno

Muestra los datos identificativos del alumno y su estado de evaluación para cada asignatura en la que está matriculado, que incluye la lista de pruebas asignadas con información de entrega y calificación, y las calificaciones en cada convocatoria.

- **Cambiar contraseña de usuario**: para cada asignatura en la que esté matriculado, si el alumno ya ha solicitado el usuario se podrá cambiar su contraseña de acceso (lo que afectará tanto al acceso a la aplicación como al acceso a la base de datos para prácticas).
- **Eliminar matrícula**: para cada asignatura en la que esté matriculado, se permitirá eliminar la matrícula siempre y cuando no haya realizado ninguna prueba ni se hayan calculado calificaciones finales para esa matrícula.
- **Matricular alumno**: permite matricular al alumno en una asignatura en la que no se encuentre matriculado previamente.

Editar alumno

Da la posibilidad de editar los datos identificativos de un alumno.

- **Eliminar alumno**, lo cual se podrá hacer sólo si no está matriculado en ninguna asignatura.

3.1.1.3 Gestión de pruebas

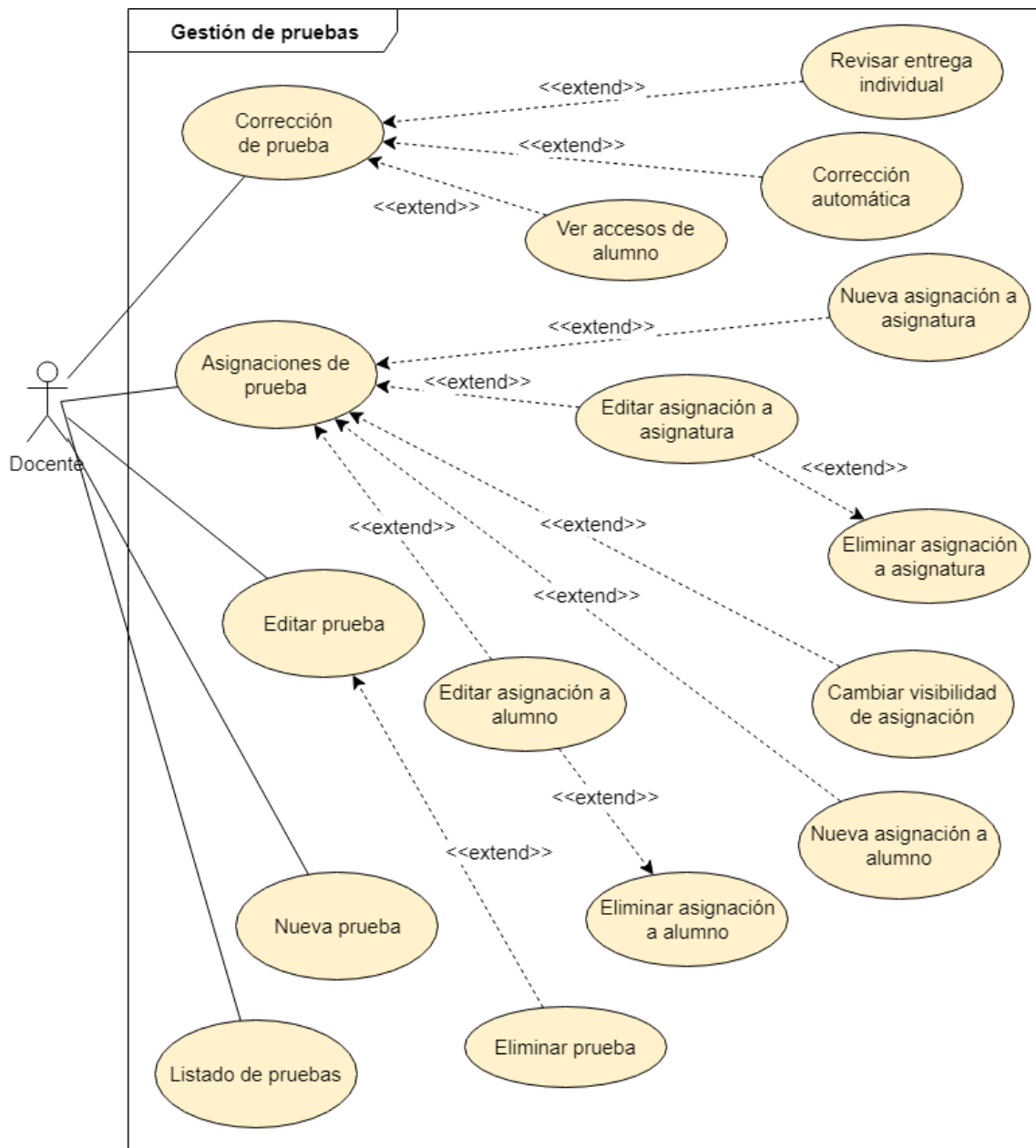


Ilustración 3.5: Diagrama de caso de uso de la gestión de pruebas

En esta sección se gestionan las pruebas que podrán ser realizadas por los alumnos desde la propia aplicación, las cuales podrán ser asignadas a distintas asignaturas. Es la parte más extensa del panel del docente.

Listado de pruebas

Obtiene una lista de pruebas en base a uno de los filtros predefinidos (todas, de un tipo en concreto, sin asignar o asignadas a una asignatura específica). Permitirá realizar búsquedas entre los resultados y, para cada asignatura, proporcionará enlaces a su edición y la gestión de sus asignaciones.

Nueva prueba

Permite crear pruebas de los dos tipos disponibles (DML y DDL) con un nombre, una descripción y un número indeterminado de preguntas. Además, para las pruebas de tipo DDL, también será necesario especificar un prefijo para los usuarios de Oracle que se crearán para cada alumno que vaya a realizarla y el nombre del procedimiento que se utilizará para la corrección automática.

Para cada pregunta se especificará la puntuación, el enunciado, el esquema del resultado (sólo para pruebas de tipo DML, indica el formato del resultado de la consulta que se pide) y una solución opcional (usada en las pruebas de tipo DML para la corrección automática).

Editar prueba

Desde donde se pueden editar los datos de una prueba, así como sus preguntas. Se pueden editar todos los datos menos el ID de la prueba, excepto cuando la asignatura ya se encuentre asignada a alguna asignatura, que no se podrá cambiar el tipo de prueba ni el número de preguntas (permitiendo sólo corregir erratas).

- **Eliminar prueba**, acción que sólo se podrá realizar en caso de que no se haya asignado a ninguna asignatura.

Asignaciones de prueba

Muestra un listado de las asignaturas, agrupadas por curso y opcionalmente por convocatoria, que tienen asignada la prueba seleccionada para que sus alumnos puedan realizarla y que forme parte de su proceso de evaluación.

Para cada asignación se muestra si está visible o no para los alumnos y la fecha de inicio y fin (entre las que los alumnos pueden realizarlas).

Además, se muestra un listado de alumnos que tienen permiso para realizar la prueba en otra fecha y hora distintas a la de la asignación a su asignatura (siendo esta requerida).

Se permite:

- **Nueva asignación a asignatura**: asignar la prueba a una asignatura a la que no se haya asignado, especificando la convocatoria, las fechas de

inicio y fin, el umbral mínimo para considerar la prueba como aprobada y una palabra de paso.

- **Editar asignación a asignatura:** editar cualquiera de los datos introducidos al crear la asignación
 - **Eliminar asignación a asignatura,** que también borrará las asignaciones extra a los alumnos de esa asignatura.
- **Cambiar visibilidad de asignación,** que permite mostrar u ocultar la prueba para los alumnos de esa asignatura, ya sea para realizarla o para ver las calificaciones (dependiendo de su estado).
- **Nueva asignación a alumno:** se puede conceder permiso a uno o varios usuarios para realizar la prueba, siempre y cuando estén matriculados en alguna asignatura que la tenga asignada.
- **Editar asignación a alumno,** permite editar la fecha de inicio y fin para las que se le permite realizar la prueba a un alumno en concreto.
 - **Eliminar asignación a alumno,** elimina la concesión del permiso para que un alumno realice la prueba en una fecha concreta.

Corrección de prueba

Para una prueba asignada a una asignatura concreta, muestra los datos de la asignación y la lista de alumnos de la asignatura, indicando para cada uno el estado de la entrega y la calificación. Se permite:

- **Ver accesos de alumno:** para cada alumno que haya accedido a la prueba, aunque no la haya entregado, se podrá ver la fecha y hora a la que accedió y la dirección IP desde donde lo hizo, junto con el código de ordenador introducido opcionalmente por el alumno.
- **Corrección automática:** corrige las entregas automáticamente mediante procedimientos almacenados de la base de datos.
- **Revisar entrega individual:** siempre y cuando el alumno haya entregado la prueba, muestra su respuesta junto con el enunciado y la solución para cada pregunta de la prueba, permitiendo introducir la nota y un comentario.

3.1.1.4 Gestión de notas externas

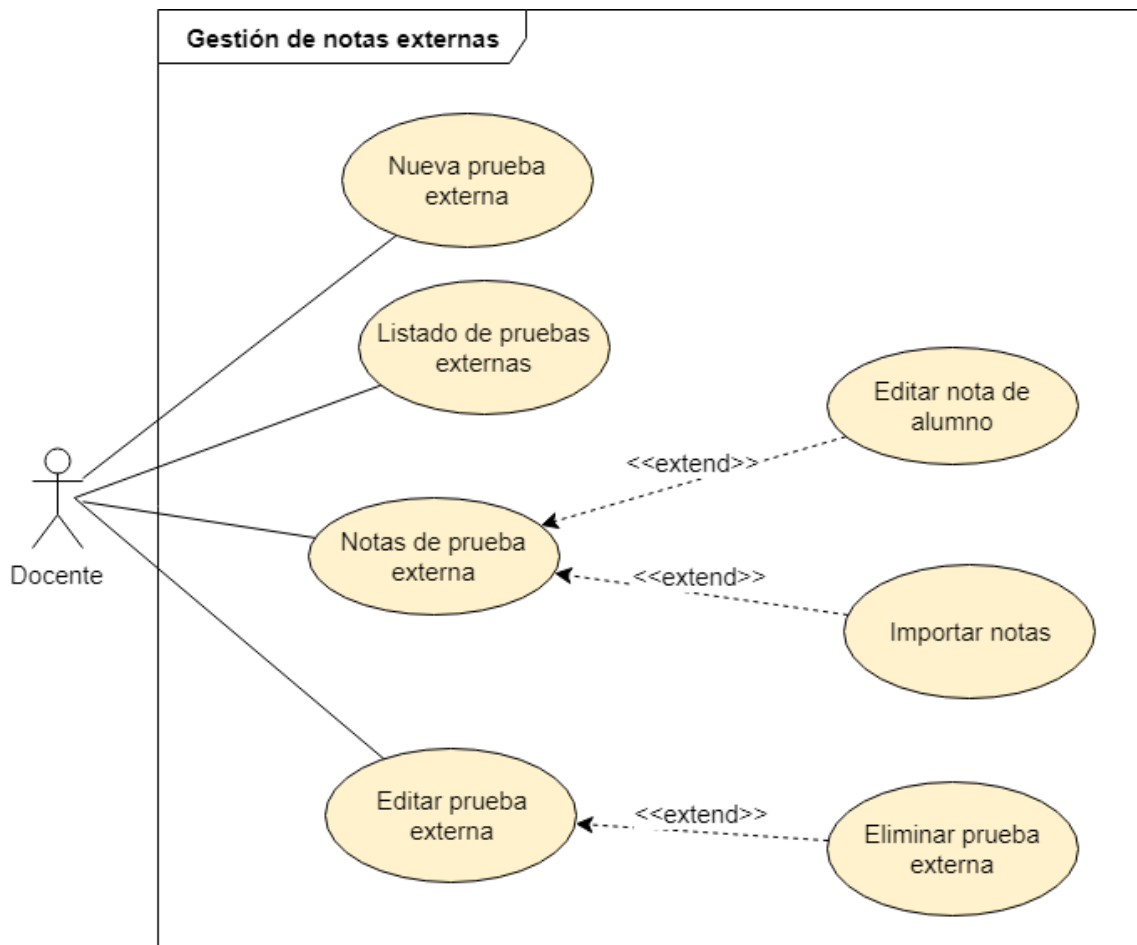


Ilustración 3.6: Diagrama de caso de uso de la gestión de notas externas

Para cada asignatura se pueden registrar las pruebas que no se han realizado desde la aplicación pero que se tienen en cuenta a la hora de generar las calificaciones finales de cada convocatoria, por lo que se pueden almacenar las notas de cada alumno de la asignatura para cada una.

Listado de pruebas externas

Muestra el listado de pruebas externas de la asignatura seleccionada con el tipo, la convocatoria y la fecha de realización.

Nueva prueba externa

Permite registrar una nueva prueba externa para una asignatura.

Editar prueba externa

Edición de la información de una prueba externa de una asignatura.

- **Eliminar prueba externa:** elimina la prueba externa de la asignatura junto con sus calificaciones.

Notas de prueba externa

Para la prueba externa seleccionada, muestra un listado de los alumnos de la asignatura, indicando para cada uno la calificación en caso de estar almacenada.

Se permite:

- **Editar nota de alumno:** fija o borra la nota de un alumno en concreto para la prueba.
- **Importar notas:** ofrece la posibilidad de subir un fichero con la lista de calificaciones de la prueba para cada alumno de la asignatura o un subconjunto de ellos. Los formatos aceptados son Excel y CSV, dando la posibilidad de desechar la primera fila y elegir si el alumno se identifica por expediente o por DNI.

3.1.1.5 Gestión de cuentas

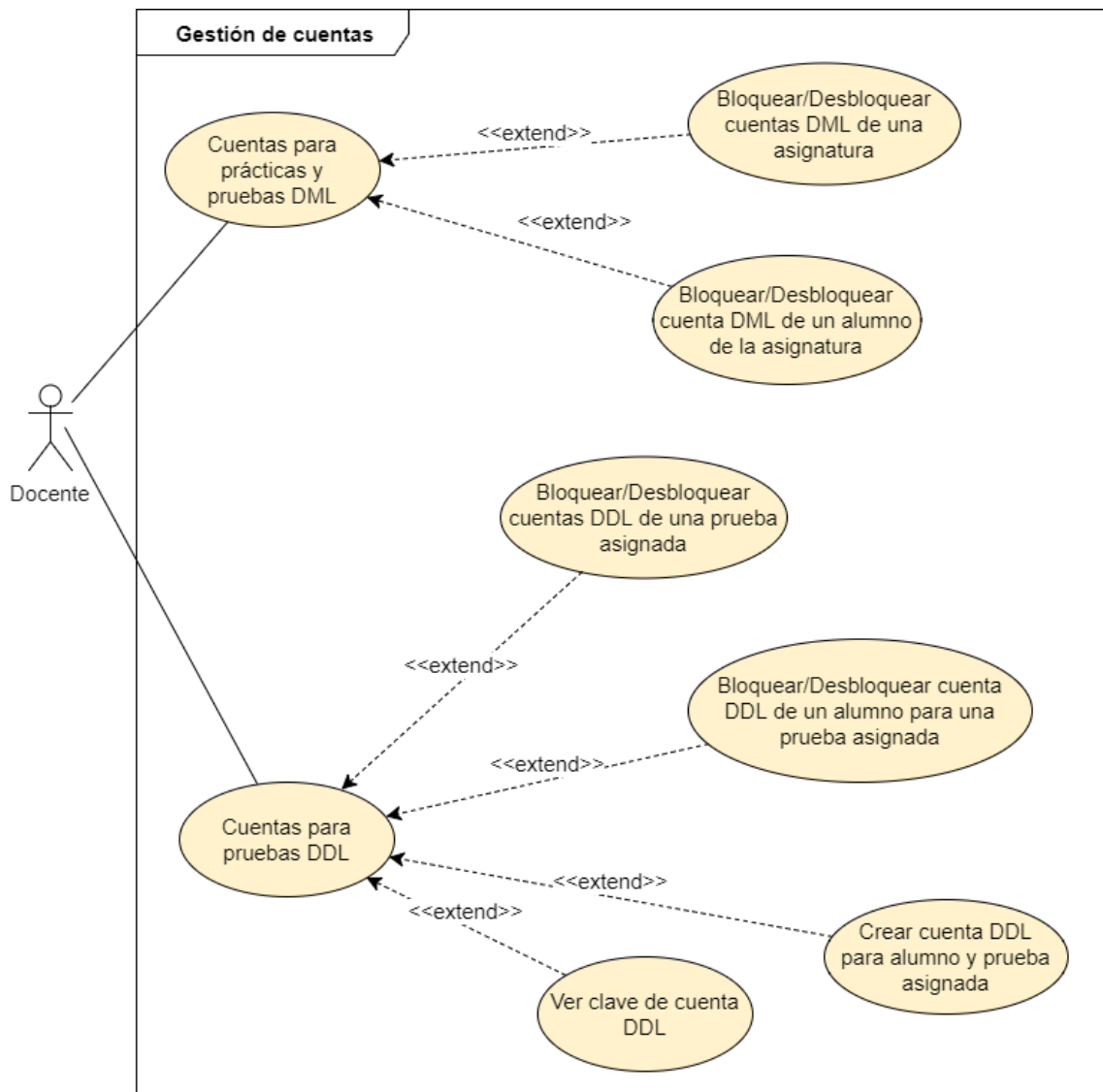


Ilustración 3.7: Diagrama de caso de uso de la gestión de cuentas

Se controlan los usuarios de Oracle en una base de datos del departamento que cada alumno utiliza para realizar los ejercicios prácticos y las pruebas evaluadas por la aplicación.

Cuentas para prácticas y pruebas DML

Estos usuarios son generados por la aplicación cuando el alumno solicita su usuario para la asignatura. Se pueden bloquear y desbloquear individualmente o por asignatura.

Cuentas para pruebas DDL

Estos usuarios son generados por la aplicación cuando se realiza la asignación de una prueba de este tipo a una asignatura. Se pueden bloquear y desbloquear individualmente o por prueba asignada y ver la clave de cada uno.

Además, se podrá crear el usuario para los alumnos que hayan sido matriculados después de asignar la prueba a la asignatura.

3.1.1.6 Informes

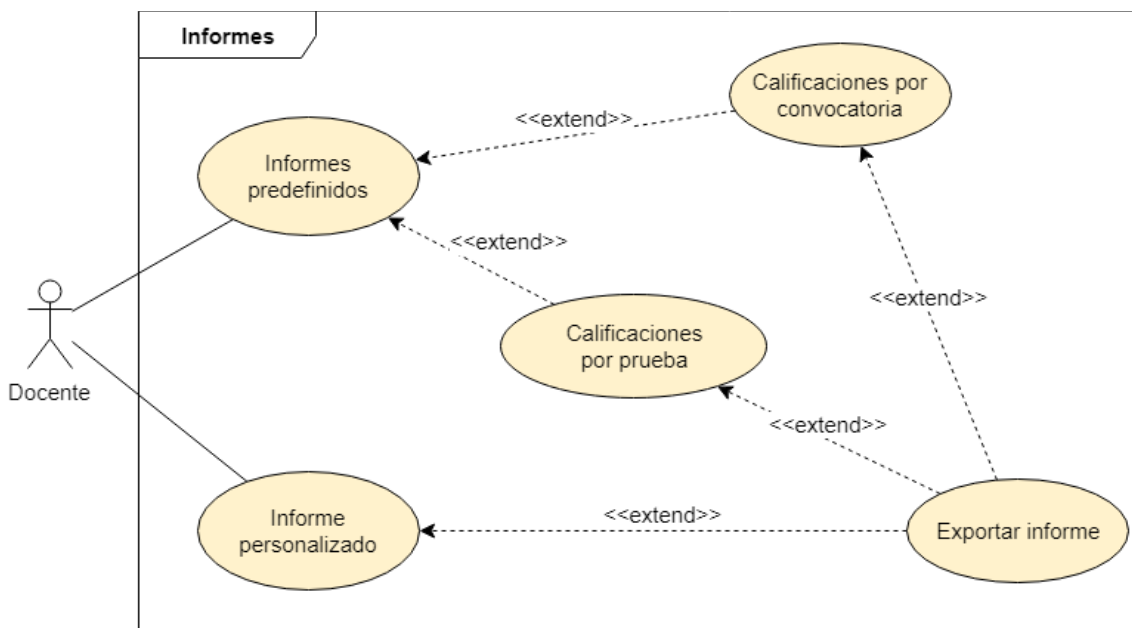


Ilustración 3.8: Diagrama de caso de uso de la generación de informes

Sección donde se permite generar informes que pueden ser exportados a formatos imprimibles o distribuibles por medios electrónicos.

Permite imprimirlo directamente o exportarlo en los formatos: PDF, Excel, CSV.

Informes predefinidos

Sólo es necesario seleccionar las entidades sobre las que se quiere obtener el informe. Por ahora sólo hay disponibles:

- **Calificaciones por prueba:** se selecciona una asignatura y una de sus pruebas asignadas y se obtiene un listado de los alumnos matriculados con la calificación de cada uno para la prueba.
- **Calificaciones por convocatoria:** se selecciona una asignatura y una convocatoria y se obtiene un listado de los alumnos matriculados con la calificación final de cada uno para esa convocatoria.

Informe personalizado

Permite obtener un informe mediante una consulta SQL de tipo SELECT, mostrando como resultado el listado de filas devueltas por la consulta con todos los campos seleccionados como columnas.

De cara a la mejor visualización o un documento exportado más presentable, permite mostrar/ocultar las columnas que se quieran de entre las que devuelve la consulta.

3.1.1.7 Configuración

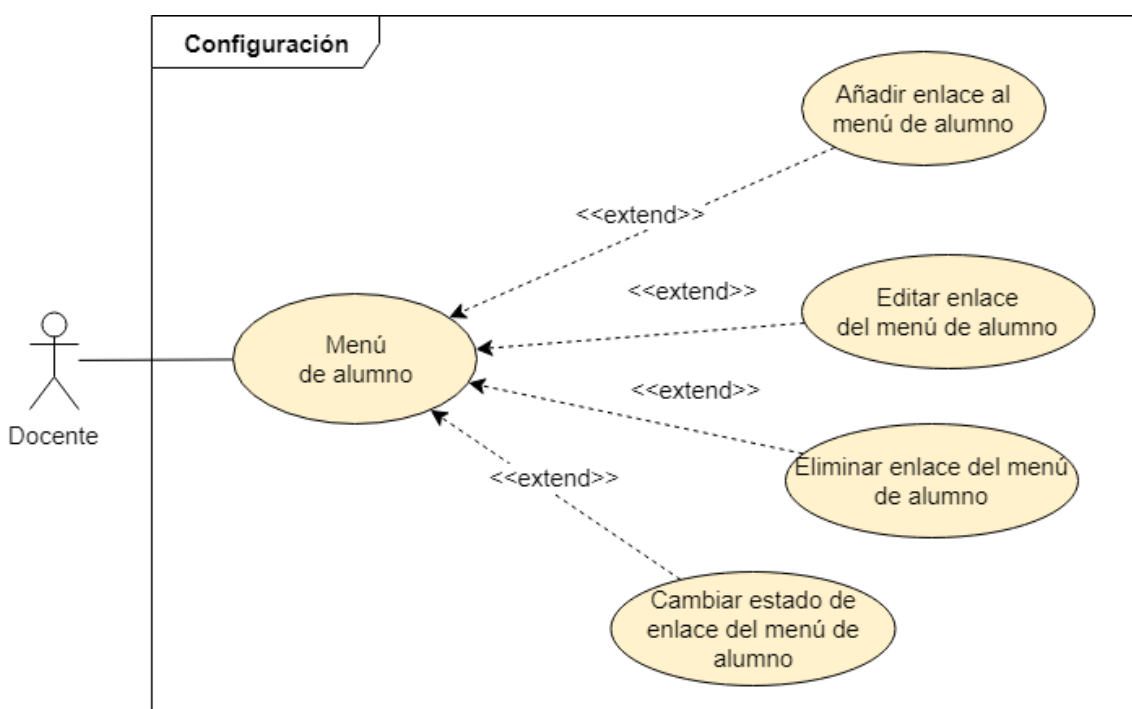


Ilustración 3.9: Diagrama de caso de uso de la configuración de la aplicación

Apartado para configurar aspectos funcionales de la aplicación.

Menú de alumno

Permite añadir enlaces externos al menú lateral que aparece a los alumnos en su panel. Se permite añadir, editar y eliminar enlaces, además de cambiar el estado de visible a oculto y viceversa.

3.1.2 Panel de alumno

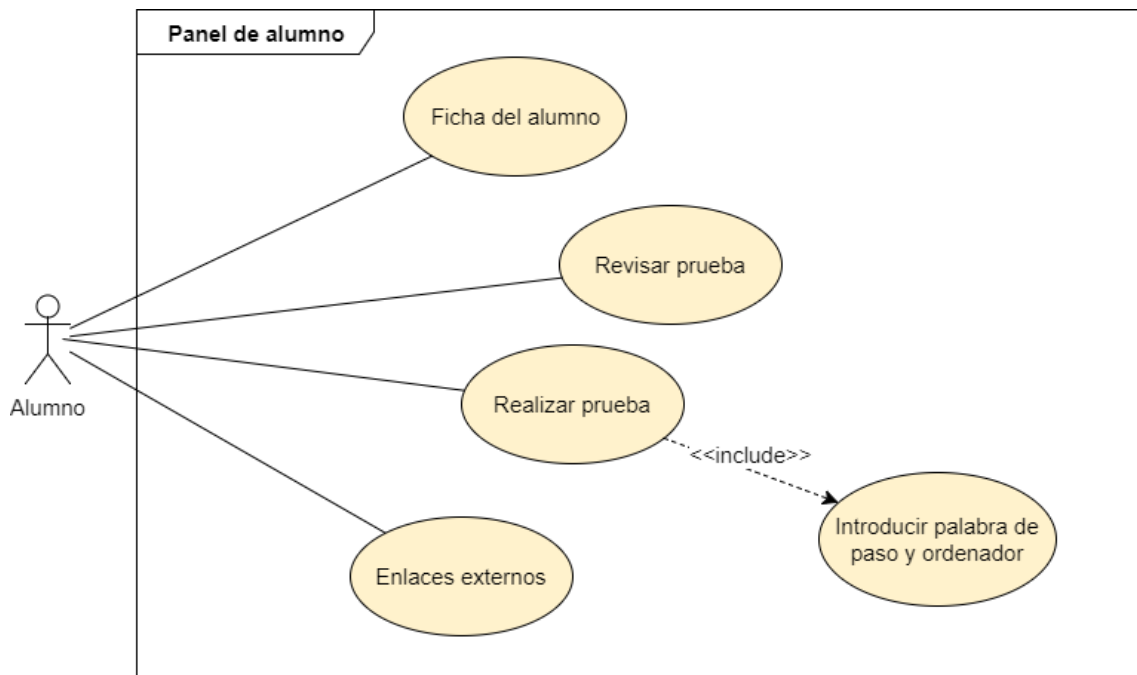


Ilustración 3.10: Diagrama de caso de uso de panel de alumno

Una vez el usuario de tipo Alumno se autentica en la aplicación, accederá a un panel más escueto que el del docente donde accederá de forma directa a las opciones que se le permiten.

Ficha del alumno

El alumno podrá ver sus datos personales almacenados en la aplicación y una lista de pruebas asignadas a la asignatura correspondiente al usuario para cada convocatoria, cada una con su estado de realización o la calificación obtenida en caso de que haya sido corregida.

Realizar prueba

El alumno podrá realizar las pruebas que estén visibles y en fecha y hora de realización (ha empezado y no ha acabado). Podrá acceder las veces que sean necesarias mientras no la entregue, siendo guardado el log de accesos.

Revisar prueba

Será posible, siempre que estén visibles, revisar la corrección de las pruebas que haya realizado el alumno una vez que, la aplicación automáticamente o el docente de forma manual, evalúe la entrega.

Enlaces externos

En el menú lateral del panel del alumno podrán aparecer enlaces externos previamente configurados por el docente.

3.2 Base de datos

La base de datos fue una de las primeras tareas a realizar en la fase de diseño de la aplicación, ya que almacena y relaciona todos los datos que se van a manejar.

En este punto se describen el conjunto de tablas y relaciones que forman el esquema de la base de datos, así como los elementos PL/SQL implementados para realizar acciones o transformaciones sobre los datos.

El motor de base de datos elegido para el desarrollo del proyecto fue Oracle, ya que es el utilizado por el departamento que usará la aplicación y el que se imparte en las asignaturas a las que va dirigida.

3.2.1 Esquema

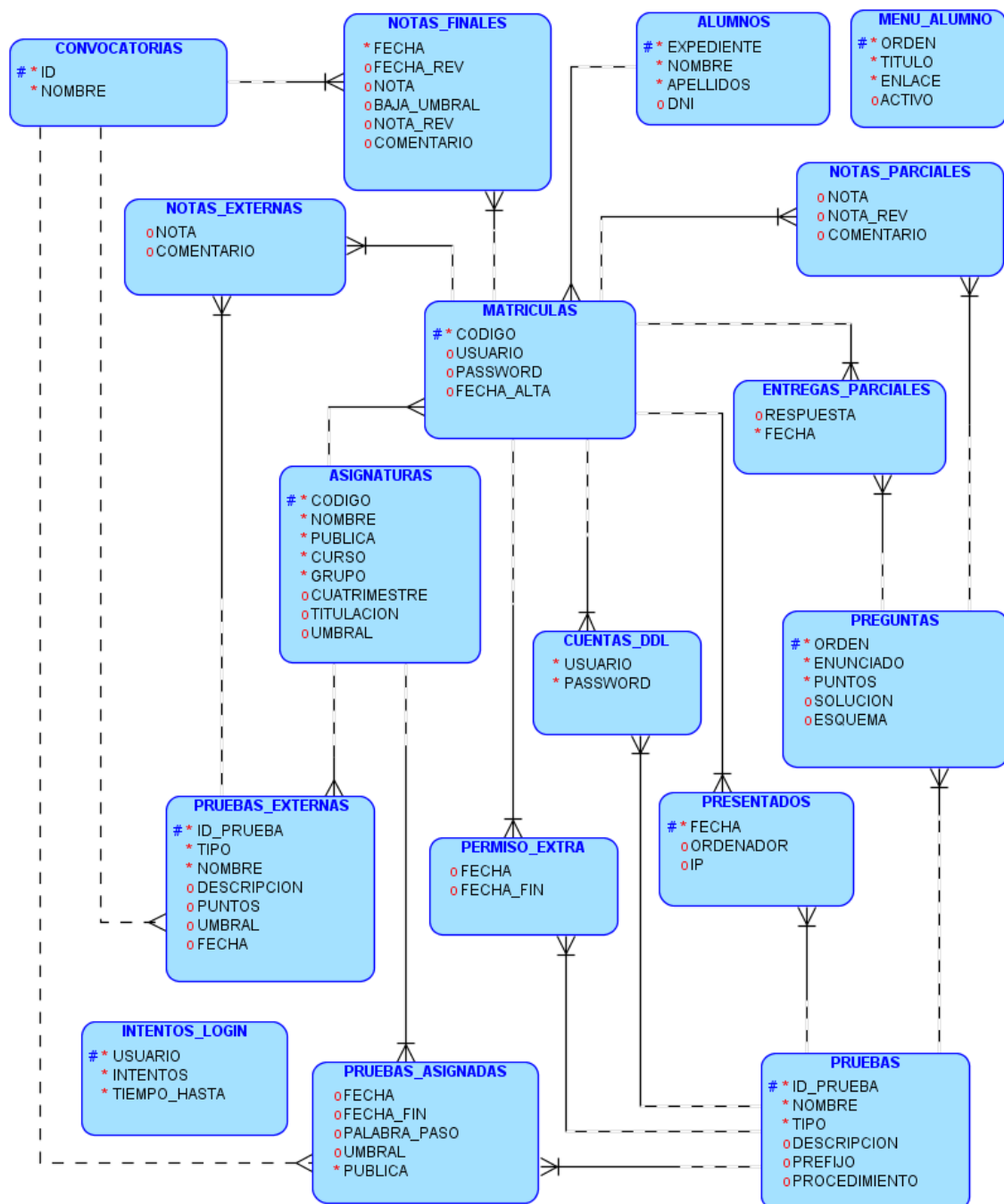


Ilustración 3.11: Diagrama Entidad/Relación de la base de datos

Tablas

CONVOCATORIAS

Almacena las distintas convocatorias disponibles para la evaluación de las asignaturas, las cuales servirán para clasificar las pruebas y notas finales.

ASIGNATURAS

Contiene información sobre las asignaturas gestionadas por la aplicación. Cada elemento se corresponde con un grupo y curso en concreto, por lo que si hay varios grupos para una asignatura existirá un elemento por cada uno que se relacionará con sus matrículas y sus asignaciones de pruebas.

ALUMNOS

Identifica de manera única a cada alumno involucrado en el proceso docente gestionado por la aplicación. Cada uno puede estar matriculado en distintas asignaturas manteniendo sus datos identificativos sin duplicar.

MATRICULAS

Una de las principales entidades del esquema, donde se almacena la relación entre cada alumno y las asignaturas en las que está matriculado. Sirve de identificación para todos los registros de evaluación y calificación.

PRUEBAS

Pruebas que el alumno puede realizar a través de la aplicación, es decir, las de tipo DML o DDL. Cuenta con dos atributos que sólo son válidos para el tipo de prueba DDL: PREFIJO, que será utilizado para formar los nombres de usuario utilizados para las cuentas Oracle con las que accederán los alumnos para realizar dichas pruebas, y PROCEDIMIENTO, que contiene el nombre del procedimiento almacenado que las corrige.

PREGUNTAS

Contiene las preguntas de cada prueba de tipo DML o DDL, identificadas por el orden de la pregunta y el ID de la prueba a la que corresponde.

PRUEBAS_ASIGNADAS

Representa la asignación de una prueba a una asignatura, permitiendo a su vez relacionarla con los alumnos matriculados en ella. Se identifica débilmente por la asignatura y la prueba, por lo que no se permite asignar la misma prueba para convocatorias distintas de una misma asignatura.

PERMISO_EXTRA

Cada elemento relaciona a un alumno matriculado con una prueba con el fin de que pueda realizar la prueba en una fecha/hora distinta a la de la asignación a la asignatura en cuestión.

CUENTAS_DDL

Guarda los datos de identificación de los usuarios que se crean para cada alumno matriculado en una asignatura a la que tenga asignada una prueba de tipo DDL.

PRESENTADOS

Registra cada vez que un alumno accede a la realización de una prueba determinada con información de tiempo y red. Con la fecha incluida en la clave se permite que un usuario inicie la prueba varias veces y con la información de red (la IP) se identifica el equipo desde donde se accedió a la prueba.

ENTREGAS_PARCIALES

Respuestas de los alumnos matriculados para cada pregunta de las pruebas que ha realizado. La matrícula solo puede hacer una única entrega de una pregunta, por lo que una matrícula puede entregar una prueba una sola vez.

NOTAS_PARCIALES

Calificaciones de cada matrícula para cada pregunta de las pruebas que ha realizado. Cada tupla contiene la nota generada automáticamente y la revisada por el profesor, prevaleciendo esta última.

PRUEBAS_EXTERNAS

Pruebas que se han realizado en una asignatura usando otro medio distinto a nuestra aplicación. De cada una se almacena solo información descriptiva y de ponderación, para poder tenerla en cuenta en las notas finales generadas.

NOTAS_EXTERNAS

Calificaciones para una prueba externa de cada alumno matriculado en la asignatura correspondiente.

NOTAS_FINALES

Calificación de cada matrícula para una convocatoria, calculadas por la aplicación en base a las calificación y puntuación de las preguntas de cada prueba realizada.

MENU_ALUMNO

Elementos adicionales del menú del alumno que enlazan a cualquier URL.

INTENTOS_LOGIN

Tabla para almacenar el número de intentos y la fecha de los intentos fallidos de autenticación como docente para cada usuario permitiendo bloquear la comprobación al alcanzar un límite determinado hasta pasado un cierto tiempo.

Secuencias

SEQ_CODIGOS_MAT y SEQ_MENU_ALUM

Generan los identificadores consecutivos para las matrículas y los elementos del menú de alumnos respectivamente.

3.2.2 Elementos PL/SQL

Triggers

PARAMETROS_NLS_SESION

Se dispara al realizar una conexión con el esquema donde se encuentra y establece la coma como separador de decimales y el punto como separador de miles, con el fin de evitar conflictos con PHP.

NUEVA_MATRICULA y NUEVO_MENU_ALUM

Se disparan cuando se inserta una nueva matrícula o un nuevo elemento del menú de alumnos respectivamente, simulando una columna autoincremental (ya que Oracle no la implementa directamente) mediante el uso de secuencias.

NUEVA_PRUEBA y NUEVA_PRUEBA_EXT

También se ejecutan al insertar una nueva fila en las tablas de pruebas y pruebas externas respectivamente, aunque sólo actúan si no se especifica identificador y le asigna el siguiente al último existente.

Funciones

VALIDA_USUARIO

Comprueba que un usuario y una contraseña corresponden con las credenciales de un usuario de Oracle.

USUARIO_ROL

Comprueba que un usuario de la base de datos tiene un rol específico concedido.

LOGIN_DOCENTE

Apoyándose en las funciones VALIDA_USUARIO y USUARIO_ROL, comprueba si un usuario y contraseña determinados corresponden a unas credenciales válidas para conectarse como docente a la aplicación.

Procedimientos

CREAR_USUARIO

Crea un usuario de Oracle con una configuración específica para que los alumnos realicen las pruebas prácticas.

BORRAR_USUARIO

Borra un usuario de Oracle.

BLOQUEAR_USUARIO

Bloquea un usuario de Oracle para que no pueda acceder a la base de datos.

DESBLOQUEAR_USUARIO

Desbloquea un usuario de la base de datos para que pueda acceder.

CAMBIAR_CLAVE_USUARIO

Cambia la clave de un usuario de Oracle.

BLOQUEAR_DML y DESBLOQUEAR_DML

Bloquea y desbloquea, respectivamente, los usuarios de Oracle que los alumnos de una asignatura en concreto utilizan para las prácticas y las pruebas de tipo DML.

BLOQUEAR_DDL y DESBLOQUEAR_DDL

Bloquea y desbloquea, respectivamente, los usuarios de Oracle que los alumnos de una asignatura en concreto utilizan para una prueba DDL específica.

CREAR_USUARIOS_DDL

Crea un usuario en la base de datos por cada alumno matriculado en una asignatura para una prueba de tipo DDL concreta.

BORRAR_USUARIOS_DDL

Borra los usuarios de Oracle de una asignatura y una prueba DDL específicas.

CORREGIR_DML

Corrige una prueba de tipo DML para los alumnos de una asignatura específica que la tenga asignada.

GENERAR_NOTAS_FINALES

Genera las notas finales de los alumnos de una asignatura para una convocatoria concreta, teniendo en cuenta las pruebas de la aplicación y las pruebas externas, así como los umbrales establecidos en la asignatura y en las asignaciones de cada prueba a la misma.

4 Implementación

La implementación de esta aplicación requiere de un nivel de una complejidad moderada de cara a la modificación y ampliación por parte de personal que no sea experto en las tecnologías utilizadas.

La solución propuesta debe aportar seguridad y efectividad en todos los casos, sin que los recursos necesarios para su ejecución sean muy elevados y siendo compatible con los medios con los que se cuenta en la Universidad.

En base a los requisitos propuestos y con el objetivo de estudiar y comprender más a fondo los pilares del desarrollo de aplicaciones web así como la interacción entre los distintos elementos que las componen, se ha decidido desarrollar un pequeño marco de desarrollo propio que sirva de base y apoyo para el desarrollo de las distintas funcionalidades requeridas.

La implementación se basa en el patrón de diseño Modelo-Vista-Controlador, utilizado en la mayoría de frameworks disponibles para el desarrollo de aplicaciones de este tipo y que permitirá tener el código bien organizado y fomentar la modularización y reutilización del mismo.

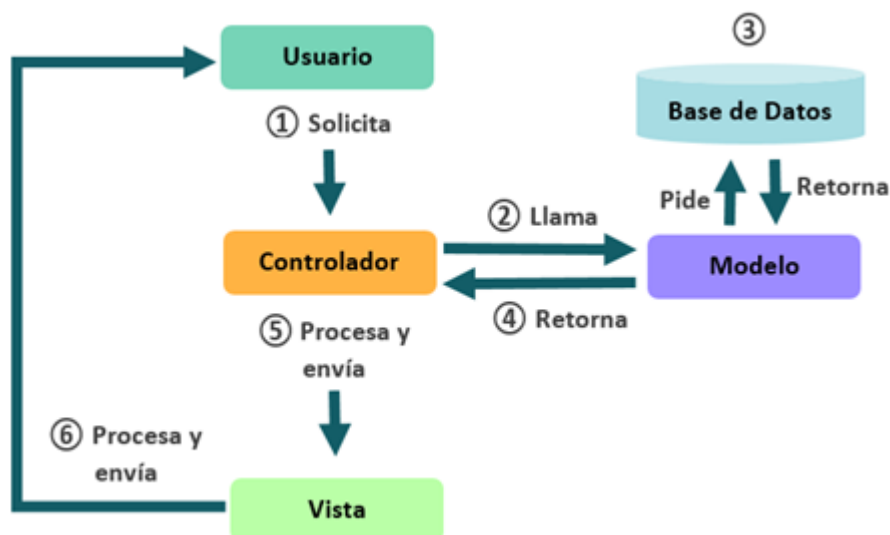


Ilustración 4.1: Patrón de diseño Modelo-Vista-Controlador

Este capítulo se centrará en la descripción técnica y funcional de la estructura y el núcleo de la aplicación, no entrando en el detalle de cada una de las secciones debido a que se reduce meramente a utilizar el lenguaje con la organización propuesta y apoyándose en las características implementadas.

Las características que incorpora el marco de desarrollo implementado son:

- Utilización de los namespaces de PHP, similares a los paquetes de otros lenguajes, para la organización de las clases de toda la aplicación
- Enrutamiento automático con URL amigables al usuario, correspondiéndose los distintos segmentos de la URI a los nombres del controlador y método de la acción ejecutada seguida de los parámetros
- Manejo de la sesión de los usuarios de la aplicación con PHP distinguiendo entre roles y permitiendo almacenar mensajes para intercambiar entre distintas peticiones
- Gestión de errores y excepciones con generación de logs detallados
- Posibilidad de manejar distintas conexiones con bases de datos Oracle
- Interfaz para simplificar las operaciones básicas con bases de datos de Oracle a través de oci8 [5]
- Posibilidad de incluir librerías externas de forma sencilla
- Mecanismos para obtención y validación de datos de entrada
- Generador y validador de tokens de cara a la protección contra CSRF

Además de los detalles de la implementación de la aplicación, en este capítulo también se especifican las labores realizadas en cuanto a la implantación de la misma en los sistemas de producción a los que va destinada así como las pruebas realizadas para comprobar la operatividad y la correcta satisfacción de los requisitos.

4.1 Metodología de desarrollo

Durante todo el proceso de desarrollo de la aplicación se ha seguido una metodología incremental iterativa, realizando iteraciones por cada parte de la aplicación pasando por las distintas fases de diseño, desarrollo y pruebas en cada una.

De esta forma cada iteración va completando el proyecto de manera que se adquiere en poco tiempo una versión final aunque no contenga todas las funcionalidades, lo que permite familiarizarse y detectar posibles cambios lo antes posible y aporta motivación al desarrollador al ver resultados.

Las principales iteraciones realizadas han sido:

- Núcleo del proyecto
- Autenticación
- Estructura del panel de docente
- Gestión de asignaturas, alumnos y matrículas desde el panel de docente
- Gestión de pruebas, asignaciones y entregas desde el panel de docente
- Corrección automática de pruebas
- Gestión de cuentas Oracle desde el panel de docente
- Panel y ficha del alumno
- Realización y revisión de prueba desde el panel de alumno
- Generación de informes desde el panel del docente
- Generar calificaciones finales por convocatoria
- Gestión de notas externas desde el panel de docente
- Añadir menú personalizado al panel de alumno y su gestión en el panel de docente

4.2 Estructura de la aplicación

En este apartado se hace un recorrido de la estructura de ficheros de la aplicación justificando su organización y detallando la función de cada una de las partes que la componen.

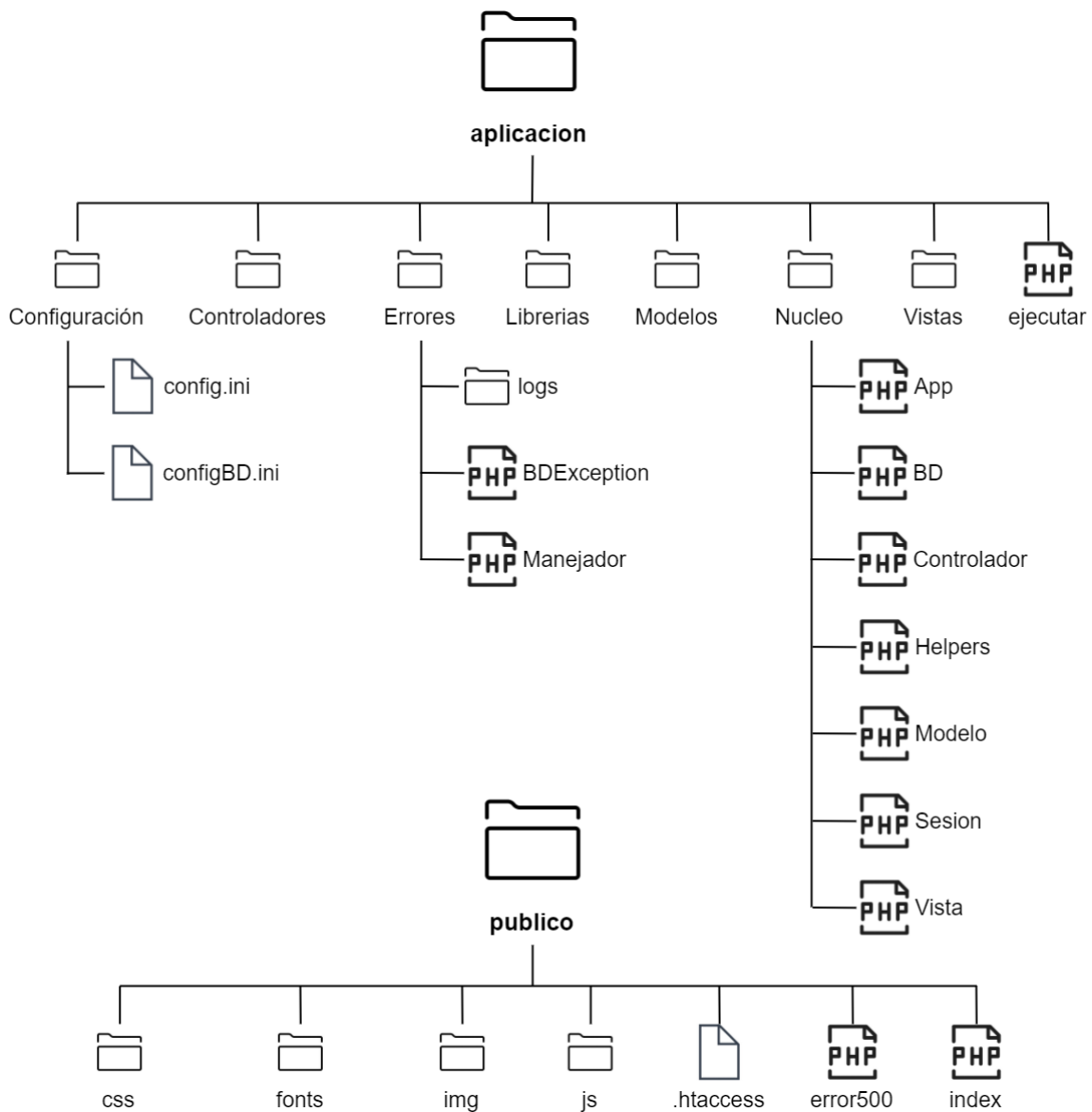


Ilustración 4.2: Jerarquía de directorios y ficheros de la aplicación

La jerarquía de directorios cuenta con dos directorios principales que son **aplicacion** y **publico**.

El directorio **aplicacion** es el que contiene los ficheros de la propia aplicación que debe procesar PHP y que no deben ser visibles ni accesibles de cara al usuario. Contiene los siguientes elementos:

- Directorio **Configuración**

Contiene ficheros con parámetros de configuración para la ejecución de la aplicación. Contiene el fichero *config.ini*, donde se guardan configuraciones generales como la URL pública de la aplicación, y el fichero *configBD.ini*, donde se guardan las credenciales y cadenas de conexión a la base de datos.

- Directorio **Controladores**

Contenedor donde se incluyen las clases que agrupan las acciones disponibles de cara al usuario, que se corresponden a cada uno de los métodos públicos que incluyen. Estas clases extenderán de la clase *Nucleo\Controlador* para adquirir las funcionalidades privadas comunes a todos los controladores.

- Directorio **Errores**

En este directorio se encuentran los elementos relacionados con la gestión de errores internos de la aplicación. En el se deben incluir las clases de los distintos tipos de excepciones, como es el caso de *BDException* para los errores relacionados con operaciones sobre la base de datos.

La clase *Manejador* contiene dos métodos que corresponden a los manejadores de errores y excepciones, que se establecen como predeterminados al iniciar la aplicación desde el fichero *ejecutar.php*.

El manejador de errores simplemente genera una excepción del tipo *ErrorException*, siendo el manejador de excepciones el encargado de generar los logs en base a los distintos tipos de excepciones generadas. En el subdirectorio *logs*, con permisos de escritura, será donde se guarden los logs generados por el manejador de excepciones.

- Directorio **Librerías**

Ubicación para colocar las librerías PHP externas para poder reaprovechar procesos y funciones concretas a la hora de implementar las funcionalidades de la aplicación. Cuando se referencia a una clase si

no se encuentra en la jerarquía de namespaces de la aplicación se buscará dentro de esta carpeta.

- Directorio **Modelos**

Engloba las clases que manejan la información con la que trabaja la aplicación, gestionando los accesos a la base de datos para obtener y guardar datos o realizar cualquier tipo de operación sobre la base de datos. Estas clases serán instanciadas y utilizadas desde los controladores y deben extender de la clase *NucleoModelo*.

- Directorio **Nucleo**

Contiene las clases y ficheros que constituyen el esqueleto de la aplicación y manejan el flujo de la misma ejecutando cada uno de los controladores que realizan las funcionalidades requeridas.

Los elementos incluidos en este directorio se detallan en un apartado de este capítulo debido a que son la base de la aplicación.

- Directorio **Vistas**

Incluye los ficheros que presentan los datos generados por los controladores incrustándolos en estructuras HTML con estilos CSS, lo cual resulta cómodo en PHP ya que permite embeberse en lenguajes de marcado. Son renderizados por la clase *Nucleo\Vista* de la aplicación.

- Fichero **ejecutar.php**

Es el fichero que ejecuta la aplicación tras preparar el entorno para la utilización de todos los componentes. Realiza la siguiente secuencia de acciones:

- Inicia la sesión de PHP del usuario
- Lee el fichero de configuración config.ini y define algunas constantes útiles para utilizarlas en distintas partes de la aplicación
- Establece una función de carga de clases automática usando namespaces, que es la forma que ofrece PHP para agrupar las clases similar a los paquetes en otros lenguajes
- Incluye el fichero Helpers.php del núcleo de la aplicación que define funciones de uso común
- Establece los manejadores de errores y excepciones y desactiva la muestra de errores por pantalla por si la configuración del servidor indicase lo contrario

- Instancia y ejecuta la aplicación, que se corresponde con la clase *NucleoVApp*

El directorio **publico** contiene los ficheros que pueden ser accesibles de cara al público ya que son recursos que se ejecutan o visualizan directamente en el lado del servidor. Los recursos se agrupan en directorios para cada tipo que coexisten con algunos ficheros necesarios para el funcionamiento de la aplicación.

- Directorio **css**
Contiene los ficheros con hojas de estilo en cascada, que se incluirán en las vistas para mejorar la apariencia de los datos presentados.
- Directorio **fonts**
Directorio con ficheros de fuentes de texto, mayormente utilizado por Bootstrap que aporta fuentes de iconos para hacer la interfaz de usuario más amigable.
- Directorio **img**
Destinado a almacenar las imágenes utilizadas por la aplicación.
- Directorio **js**
Alberga los ficheros con código Javascript, que es el utilizado para realizar acciones dinámicas sobre la aplicación desde el navegador del usuario.
- Fichero **.htaccess**
Es un archivo de configuración soportado por los servidores web Apache que nos permite en este caso traducir direcciones URL amigables que no correspondan a recursos del directorio publico a otras que pueda procesar nuestra aplicación, redirigiéndolas al fichero **index.php** pasándole como parámetro GET la URI amigable de forma transparente al usuario.
- Fichero **error500.php**
Vista que se muestra cuando ocurre un error interno en la aplicación que no se ha sido capturado por el flujo de la aplicación.
- Fichero **index.php**
Fichero de entrada de todas las peticiones a la aplicación y cuya única función es ejecutar el archivo **aplicación/ejecutar.php** para iniciar la aplicación.

4.3 Núcleo de la aplicación

La parte esencial de la aplicación se corresponde con el núcleo, que contiene el código necesario para la ejecución del flujo de la aplicación y sirve de base y apoyo para el desarrollo de las distintas funcionalidades que se permiten al usuario.

Está formado por un conjunto de clases de distinto tipo y con distintos fines las cuales se recogen en el siguiente diagrama de clases.

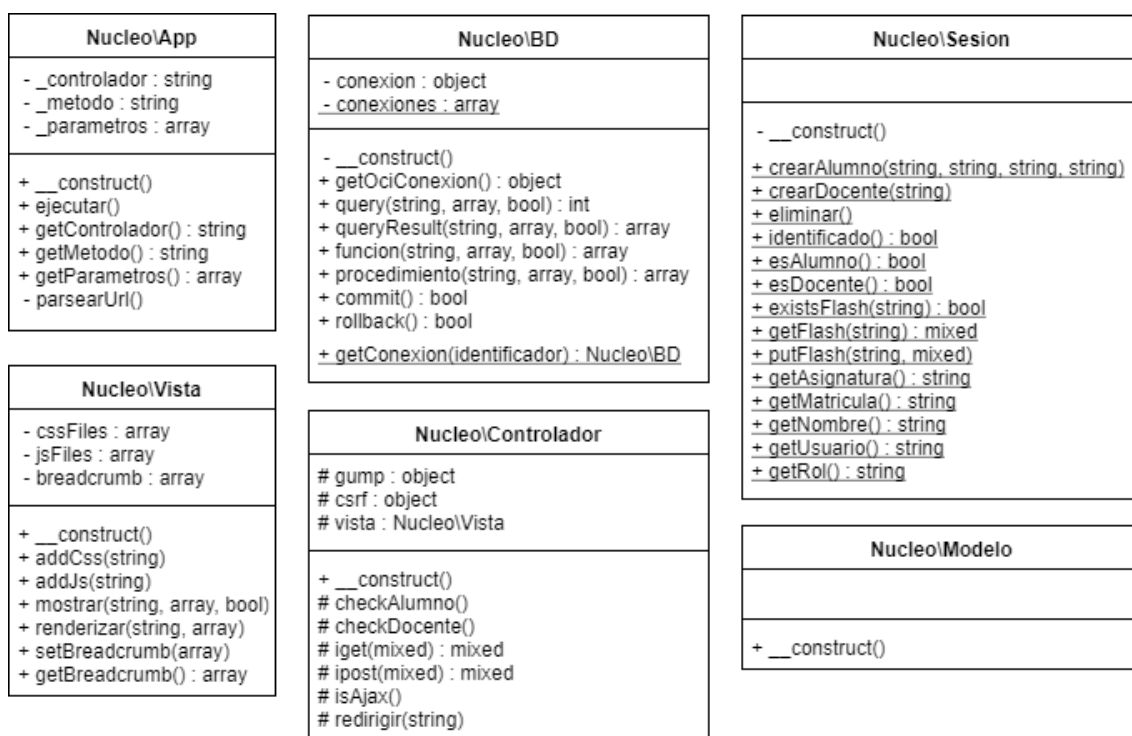


Ilustración 4.3: Diagrama de clases del núcleo de la aplicación

A continuación, procedemos a describir cada una de estas clases, con las que será necesario estar familiarizado para añadir nuevas funcionalidades a la aplicación o editar las ya existentes.

Clase App

Es la clase principal de la aplicación y controla el flujo interno para cada una de las distintas peticiones, siendo la encargada de parsear la ruta mediante la que accede el usuario para determinar qué método y controlador debe ejecutar.

La tarea de parseo de la URI se realiza al instanciar la clase, que instancia el controlador y almacena los parámetros en sus propiedades, generando una redirección a la página principal en caso de no encontrarse el controlador y método indicados.

La correspondencia entre la URI y el método a ejecutar se determina de la siguiente forma:

- Los primeros segmentos de la URI se corresponderán con los subdirectorios dentro de la carpeta Controladores y el nombre del controlador, prevaleciendo el fichero ante el directorio
- El siguiente segmento se corresponderá con el método a ejecutar
- El resto de segmentos serán pasados como argumentos al método en cuestión

Para iniciar continuar con el flujo de la aplicación una vez instanciada la clase sólo es necesario utilizar el método *ejecutar*, que ejecuta el método de la instancia almacenada mediante la función *call_user_func_array* de PHP.

Clase BD

Gestiona las conexiones con bases de datos Oracle y proporciona una interfaz de funciones que simplifican las operaciones utilizando la extensión de PHP oci8.

Sigue el patrón de diseño Factory Method, que consiste en que sólo se permite generar instancias de la misma a partir de un método estático destinado a ello. De esta forma se controla la reutilización de conexiones para un mismo identificador.

En este caso este método aceptará un parámetro opcional con el identificador de la conexión en el fichero aplicación/Configuracion/configBD.php, devolviendo una instancia de la clase con el manejador de conexión de oci8 inyectado.

La clase instanciada permite obtener el manejador de conexión de oci8 mediante la función *getOciConexion*, además de ofrecer las siguientes funciones para realizar operaciones de forma sencilla utilizando la técnica de vinculación de parámetros:

- Las funciones *funcion* y *procedimiento* ejecutan los elementos que su nombre indican
- Las funciones *query* y *queryResult*, ejecutan consultas a la base de datos que no devuelve o de vuelven datos, respectivamente siendo el valor de retorno de la primera será el número de filas afectadas por la operación
- Las funciones *commit* y *rollback* permiten manejar las transacciones en caso de haber ejecutado las funciones anteriores con el autocommit deshabilitado (último parámetro booleano de dichas funciones)

Clase Controlador

Clase genérica de la que deben extender todos los controladores y que incluye funciones para operaciones comunes a este tipo de elemento apoyándose de algunas librerías que son inyectadas en el constructor de la clase.

- Las funciones *ipost* y *iget* permiten obtener parámetros de las peticiones de tipo POST y GET respectivamente, permitiendo obtener uno o varios en función de si se le pasa una cadena o un array como argumento
- Las funciones *checkAlumno* y *checkDocente* comprueban si el usuario está autenticado con el rol correspondiente y detiene el flujo de ejecución en caso contrario, redirigiéndolo a la página principal
- La función *redirigir* redirige al usuario a la URL pasada por parámetro, aceptando una dirección relativa para dentro de la aplicación
- En la propiedad *gump* se inyecta una instancia de la librería utilizada para validar los datos de entrada de modo que se pueda utilizar de manera directa desde el código de los controladores
- En la propiedad *csrf* se inyecta una instancia de la librería utilizada para generar y validar tokens de cara a la protección contra CSRF
- Se crea una instancia de la clase *Nucleo\Vista*, que permitirá al controlador renderizar las vistas de la aplicación con los datos procesados de manera sencilla y directa.

Clase Modelo

Clase genérica de la que deben extender los modelos de la aplicación para mantener el tipado de cada componente y poder compartir funciones o comportamientos heredados entre ellos en un futuro.

Clase Sesion

Clase estática (todos sus métodos son estáticos y se permite instanciarla) que permite la gestión completa de la sesión del usuario desde cualquier parte de la aplicación, con funciones para poder llevar a cabo la autenticación para los dos roles distinguidos.

Además de la autenticación, se han añadido las funciones *put*, *get* y *exists* para el manejo de datos entre distintas páginas que serán eliminados automáticamente una vez consultados. Esto ayuda a la implementación de mensajes de error como respuesta de las acciones del usuario.

Clase Vista

Permite la renderización de las vistas ubicadas en el directorio aplicación\Vistas en base los datos generados por la aplicación.

La renderización funciona de tal manera que se ejecuta la vista elegida dentro de un contexto en el que existen variables para cada elemento del array de datos pasado con el índice de nombre y su valor correspondiente. Esto se hace mediante el método *renderizar* aprovechando las herramientas para el manejo de buffers que aporta PHP.

Facilita la integración de nuevas vistas con la aplicación, permitiendo incluirlas en el marco de diseño de la aplicación en función del rol del usuario automáticamente mediante el método *mostrar*. A este esqueleto de la aplicación es posible añadir dependencias css y js mediante los métodos *addCss* y *addJs*.

Además de las clases anteriores, en el núcleo también existe un fichero llamado **Helpers**, en el que simplemente se declaran funciones comunes a todo el proyecto (se pueden utilizar desde cualquier parte del código) para evitar repetir el código de acciones utilizadas frecuentemente.

Este fichero lo ejecuta el encargado de iniciar la aplicación, que lo incluye antes de ejecutar la aplicación y en el namespace raíz para que este accesible globalmente.

4.4 Seguridad

Uno de los requisitos más importantes para este proyecto es la seguridad, puesto que se trata con información confidencial como son los datos personales de los alumnos y sus datos académicos.

A continuación, se detallan las consideraciones de seguridad que se han tenido en cuenta en el desarrollo de la aplicación y que son hoy en día prácticamente imprescindibles a la hora de desarrollar aplicaciones que puedan ser accesibles por un conjunto no muy cerrado de usuarios.

Claves de usuarios encriptadas

Las claves de los alumnos son guardadas en base de datos encriptadas con un algoritmo de hash de un solo sentido para que resulte imposible ver cual contraseña introdujo el usuario. Para ello se ha utilizado el algoritmo Blowfish que proporciona hashes seguros y de forma rápida.

De esta manera se mantiene la privacidad del alumno y se evita que los datos de autenticación sean comprometidos por una fuga de datos o acceso malintencionado a la base de datos.

Protección contra CSRF

Este tipo de ataques ocurre cuando un sitio web que está abierto en el mismo navegador se aprovecha de la sesión que tiene abierta el usuario con la aplicación al estar autenticado para realizar peticiones que hagan acciones no autorizadas. Para prevenirlo:

- Todas las peticiones aceptadas que conlleven acciones o cambios sobre los datos de la aplicación utilizarán el método POST. Evitamos las peticiones de tipo GET ya que incluyen los parámetros en la propia URL que puede ser manipulada y referenciada desde otros sitios fácilmente.
- Se incluye un token en el formulario previo a una acción y en la sesión del usuario para compararlos cuando se intenta realizar la acción, obligando al usuario a pasar por el formulario antes de realizar la acción.

Protección contra XSS

Esta vulnerabilidad consiste en introducir código malicioso y que se ejecute en la aplicación para la sesión actual o incluso de forma permanente y para otros usuarios.

Esto ocurre cuando las entradas de datos de la aplicación no son filtradas y sanitizadas correctamente, y para solucionarlo se utiliza una librería de validación y filtrado de entrada para PHP como es GUMP.

Protección contra SQL Injection

Este ataque se ejecuta de forma semejante al anterior, concretamente introduciendo código SQL o caracteres que alteran las consultas realizadas de forma insegura.

Para prevenirlo, además de filtrar todas las entradas convenientemente como en el punto anterior, se utiliza la técnica de vinculación de parámetros, por la cual en el texto de la consulta se incluyen comodines y los datos se pasan como parámetros para ser sustituidos internamente sin que la consulta cambie.

Uso exclusivo de conexiones seguras

Para prevenir ataques de red como el de “Hombre en medio”, ya que si el tráfico entre el cliente y el servidor web no está cifrado es posible que alguien que esté conectado a la misma red pueda leer o modificar la comunicación entre ellos.

Para que todas las conexiones sean seguras sólo se permite la conexión con el servidor web bajo HTTPS, que cifra la información mediante SSL con el certificado instalado en el servidor y que ha proporcionado la autoridad certificadora que respalda los dominios de la universidad.

Configuraciones de seguridad adicionales en el servidor

Además de la aplicación, una de las cosas que se ha realizado con el proyecto ha sido la configuración del servidor en el que estará alojada. Por ello es conveniente comentar también las nociones de seguridad aplicadas, que básicamente han sido:

- Se ha instalado un cortafuegos para denegar el acceso no autorizado y permitir sólo comunicaciones por los puertos correspondientes a los servicios específicos que son utilizados, que en este caso son HTTP (aunque sólo para redirigir a la versión HTTPS), HTTPS y SSH.
- Se ha creado un usuario con privilegios temporales a través del comando `sudo`, para evitar el uso del usuario `root` que tiene todos los privilegios y puede desencadenar en acciones equivocadas irremediables o que se comprometa el servidor.
- CentOS incluye el módulo de seguridad para el núcleo de Linux llamado SELinux (Security-Enhanced Linux), el cual proporciona una política de control de accesos sobre los recursos de la máquina.

Bloqueo de usuario tras múltiples intentos de autenticación

Para evitar ataques de fuerza bruta contra la autenticación de la aplicación se guarda información en base de datos de los intentos fallidos por usuario, permitiendo limitar el número de comprobaciones en un intervalo de tiempo.

De este modo reducimos la probabilidad de que un atacante, en conocimiento de parte de la clave o a causa de una contraseña débil, consiga hacerse con la contraseña a base de intentos reiterados.

4.5 Aspecto visual e interacción con el usuario

En cuanto a la presentación de la aplicación cabe destacar algunas características que aporta la utilización de un framework de lado del cliente como Bootstrap con la ayuda de la librería jQuery, siendo las principales:

- Compatibilidad con todos los navegadores relativamente modernos, proporcionando una capa de abstracción sobre las funciones y estilos Javascript y CSS que funcionan de forma diferente en distintos navegadores.
- Diseño responsivo, lo cual quiere decir que se ve de forma adecuada en dispositivos de distintos tamaños, gracias al sistema de capas en rejilla y las *Media Queries* que permiten organizar y mostrar elementos en base a las dimensiones de la ventana.

- Componentes visuales por defecto que mejoran mucho la usabilidad de la aplicación y la experiencia del usuario, como botones, menús desplegables, paginaciones, alertas...
- Diseño y control de ventanas modales que son útiles para ver el detalle de alguna entidad o realizar acciones puntuales sin refrescar el contenido actual en pantalla.
- Paleta de colores agradables y estilos para presentar y organizar mejor el contenido.

Para construir la interfaz se han utilizado colores básicos y cómodos para una utilización prolongada de la aplicación, manteniendo un diseño sencillo e intuitivo para el usuario.

Docencia de Bases de Datos
Departamento de lenguajes y ciencias de la computación

Identificarse

Usuario

Contraseña

☒ Alumno ☐ Profesor

Conectar

[Solicitar usuario - Recordar usuario](#)

Ilustración 4.4: Captura de la pantalla de autenticación

Docencia de Bases de Datos
Departamento de lenguajes y ciencias de la computación

prueba Docente Salir


Panel docente / Gestión de asignaturas

Curso: 18/19

Buscar:

#	Nombre	Grupo	Cuat.	Pública
Ingeniería del software				
AAA	Administración Acciones Acid	A	1	Sí
BDB	Bases de Datos	B	1	Sí
Ingeniería informática				
ABDA	Administración de base de datos	A	2	Sí

Ilustración 4.5: Captura de la pantalla de asignaturas del panel de docente



Docencia de Bases de Datos
Departamento de lenguajes y ciencias de la computación

Grey Akast Enos
Bases de Datos

Salir

- Panel alumno
- Campus Virtual
- Manual SQL
- Configuración BD

Ficha Realizar prueba Revisar prueba

Datos personales

Expediente 1234697

DNI 77785676R

Nombre Grey

Apellidos Akast Enos

Datos de la asignatura

Nombre Bases de Datos

Grupo B **Curso** 18/19

Titulación Ingeniería del software

Cuatrimestre Primero

Pruebas asignadas

Febrero / Junio Septiembre Diciembre Sin convocatoria

Nombre	Tipo	Entrega	Calificación	
Prueba DDL Noviembre	DDL	No entregado	No corregido	
Primera prueba DML	DML	22/11/2018 09:21	3,83	Revisar
Para practicar DML	DML	18/10/2018 14:19	No corregido	
Simulacro DML	DML	29/09/2018 11:32	1,45	Revisar

Ilustración 4.6: Captura de pantalla de la ficha del alumno

Para enriquecer la interacción con el usuario y mejorar la usabilidad aprovechando la dinamicidad que aporta la programación de la do del cliente, se han utilizado algunas técnicas y librerías que complementan a Bootstrap y jQuery, entre las que se pueden resaltar:

- Utilización de AJAX para carga de contenido y realización de acciones de forma asíncrona sin que sea necesario recargar la página en la que se encuentra el usuario. Es una característica que ofrece Javascript y su implementación es mucho más sencilla utilizando jQuery.
- Se incorpora la librería Datatables, una extensión de jQuery para dotar a las tablas de la aplicación de características como paginado, búsqueda, ordenación por columnas, entre otras muchas posibles configuraciones. Además de las opciones por defecto, se han añadido plugins para mejorar el responsive y para permitir imprimir y exportar a ficheros CSV, Excel o PDF los datos de las tablas.
- Bootstrap cuenta con infinidad de extensiones para añadir nuevos componentes y funcionalidades a la interfaz, habiéndose utilizado algunas

como campos desplegables de selección con buscador, campos de selección de fecha y hora con interfaz de calendario o botones de estado con función de activación/desactivación.

- Editor WYSIWYG para permitir al usuario aplicar estilos y estructuras para los campos de texto de los enunciados de las preguntas, permitiendo añadir imágenes, enlaces y resaltar datos importantes. Para ello se ha utilizado nicEdit, que es una librería Javascript sencilla y ligera que permite obtener el código HTML generado fácilmente.

4.6 Configuración de la base de datos

Para el correcto funcionamiento de la aplicación y obtener el mayor rendimiento posible en la conexión entre el servidor web y la base de datos se han realizado algunas operaciones del ámbito de la administración de bases de datos.

- Se ha establecido un disparador que se ejecuta al inicio de cada sesión para establecer los caracteres de separación de miles y decimales, evitando conflictos con PHP en caso de que la base de datos tenga una configuración distinta.
- Para todas las tablas del esquema de la base de datos se han renombrado incluyendo un prefijo y se han generado vistas para cada una de ellas, de modo que se puedan hacer cambios en la estructura de la base de datos sin afectar al funcionamiento de la aplicación. Para hacerlo de forma automática, se han elaborado sentencias que generan el listado de consultas necesarias apoyándose en el diccionario de la base de datos.
- Ha sido necesario aumentar el número de sesiones concurrentes que el usuario que contiene el esquema de la aplicación puede soportar. Para ello se modifica el límite SESSIONS_PER_USER del perfil asociado al usuario.
- Para mejorar el aprovechamiento de los recursos de la máquina donde reside la base de datos utilizadas en aplicaciones que realizan muchas conexiones de corta duración, como es el caso de las aplicaciones web, se ha utilizado una característica de Oracle llamada DRCP (agrupamiento de conexiones residentes de base de datos).

Esta característica permite que se compartan procesos de la base de datos entre distintas conexiones desde la aplicación manteniendo un grupo de procesos escalable en vez de que se cree un proceso único y se termine para cada petición a la aplicación que lo requiera.

- Se han utilizado funciones criptográficas y el acceso a tablas restringidas del diccionario de datos de Oracle implementar funciones que permitan realizar la autenticación del docente a través de usuarios de la propia base de datos con un cierto rol.

Las sentencias que generan las consultas para renombrar las tablas y generar las vistas con sus respectivos nombres son las siguientes:

```
-- Renombra todas las tablas del usuario añadiéndoles un prefijo
select 'RENAME TABLE '||table_name||' TO T_'||table_name||';'
from all_tab_columns ac join all_objects ao ON (ac.owner = ao.owner
        AND ac.table_name = ao.object_name)
where object_type = 'TABLE' AND ac.owner = 'TFG_GAMARRO'
group by table_name;

-- Crea una vista para cada tabla con el mismo nombre sin el prefijo
añadido
select 'CREATE VIEW '||table_name||' ('
|| LISTAGG(column_name, ', ' ) WITHIN group(ORDER BY table_name)
||') AS SELECT '
|| LISTAGG(column_name, ', ' ) WITHIN group(ORDER BY table_name)
||' FROM T_'||TABLE_NAME||';'
from all_tab_columns where owner = 'USUARIO' group by table_name;
```

Para cambiar el límite de sesiones de perfil del usuario se utiliza la sentencia:

```
ALTER PROFILE NOMBRE_PERFIL LIMIT SESSIONS_PER_USER 200;
```

En cuanto a la característica DRCP de Oracle, para utilizarla solo hay que ejecutar en la base de datos, con permisos de administrador, la sentencia:

```
execute dbms_connection_pool.start_pool;
```

Con eso quedará activa la característica y se mantendrá un pool de conexiones. A la cadena de conexión utilizada por la aplicación para conectarse a la base de datos habrá que añadir el parámetro '(SERVER=POOLED)' en caso de utilizar la nomenclatura larga o ':POOLED' en caso de utilizar la nomenclatura corta.

En la siguiente imagen extraída de la documentación de Oracle [6] se ilustra la diferencia entre no usar y usar DRCP para las conexiones a la base de datos, donde se puede observar que sin utilizarla quedan muchos procesos desaprovechados debido a la brevedad de las transacciones que realizan.

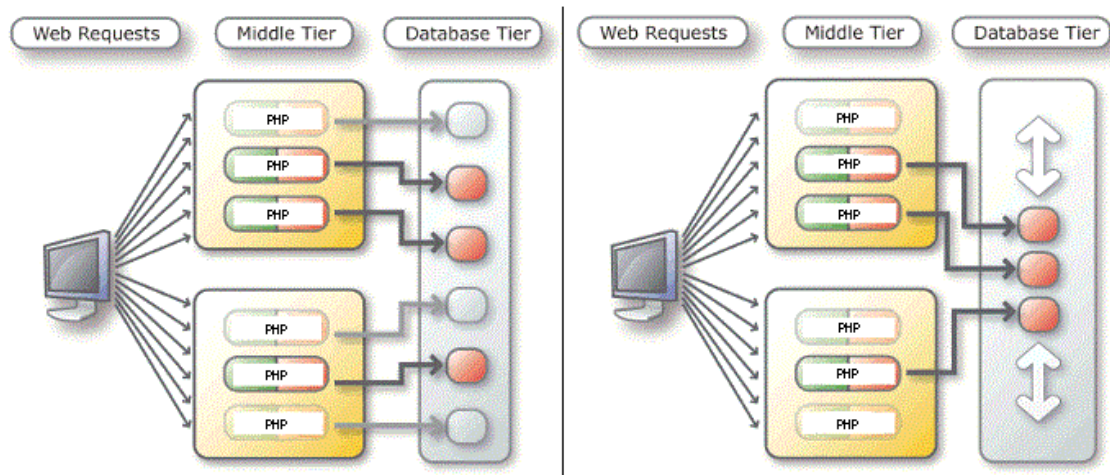


Ilustración 4.7: Conexiones a la base de datos sin DRCP y con DRCP

4.7 Infraestructura

Para alojar la aplicación se decidió utilizar una máquina virtual alojada en uno de los hipervisores de la universidad, ya que la aplicación no requiere de gran capacidad de cómputo para ejecutarse y usar una máquina dedicada para ello sería desaprovechar los recursos.

Los requisitos principales son la capacidad de conectarse a bases de datos Oracle alojadas en otros servidores de la universidad y servir la aplicación desarrollada en PHP de forma segura.

Como sistema operativo se ha utilizado **CentOS 7**, en el que se ha instalado:

- **VMware Tools** para sacar el máximo rendimiento de la máquina en el hipervisor ESXi. Esto son un conjunto de drivers
- **Apache** como servidor web para servir la aplicación a los usuarios.
- **Oracle InstantClient** para conectarse a bases de datos Oracle remotas
- **PHP** para que Apache interprete el código de nuestra aplicación, junto con la extensión oci8 para realizar consultas a la base de datos desde la aplicación.

Además, se ha configurado:

- **Firewalld** (incluido en CentOS por defecto) como cortafuegos para permitir conexiones al servidor sólo mediante los protocolos SSH, HTTP y HTTPS.
- **Servidor SFTP** (FTP bajo SSH) enjaulado en el directorio que contiene todos los ficheros de la aplicación. Todos los usuarios pertenecientes a un determinado grupo se le permitirá acceder por SFTP y sólo podrá ver los ficheros de la aplicación, evitando así utilizar usuarios con más privilegios que puedan comprometer la máquina.
- **Certificado SSL** en Apache para permitir conexiones seguras a la aplicación mediante HTTPS. El certificado ha sido proporcionado por el departamento correspondiente de la Universidad de Málaga.
- **Virtualhosts** de Apache de manera que la aplicación sólo sea accesible por HTTPS, haciendo una redirección a este protocolo en caso de acceder por HTTP.

Los pasos a seguir para la configuración del servidor se indican en el Anexo 7.4 de este documento.

4.8 Pruebas

4.8.1 Pruebas funcionales

En cada iteración de la implementación del proyecto se han realizado pruebas funcionales para cerciorarnos de que todo funciona conforme a lo estipulado en la fase de diseño.

En algunas iteraciones ha sido necesario repetir pruebas correspondientes a otras por estar relacionadas en cuanto a funcionalidad o por compartir funciones que han sido modificadas.

A continuación, se enumeran las pruebas realizadas para cada iteración del proyecto, que se corresponden con las distintas secciones o grupo de funcionalidades de la aplicación. Todos los casos de prueba han sido revisados y en caso de no resultar correcto se han realizado los cambios necesarios para cumplirlos.

Debido a la extensión de la funcionalidad de la aplicación solo se indican las condiciones a cumplir para las pruebas específicas de cada iteración y se recogen en una primera tabla las pruebas comunes a todas las iteraciones para que no resulte repetitivo.

1. Casos de prueba comunes	
ID	Condición a cumplir
1.1	Las acciones solo las puede realizar un usuario autenticado con el rol que le corresponda
1.2	Cuando se realizan las acciones se ven reflejados los cambios en la aplicación
1.3	Se muestran alertas de errores de validación al introducir datos incorrectos en formularios
1.4	Cuando hay errores de validación se vuelve al formulario con los datos introducidos
1.5	Se muestran mensajes de notificación al realizar acciones sobre los datos satisfactoriamente
1.6	Se requiere la introducción de los campos obligatorios de los formularios para su envío
1.7	Los campos opcionales se pueden dejar vacíos
1.8	La búsqueda y ordenación funcionan correctamente en las tablas que se permita
1.9	En los listados con posibilidad de filtrado se obtienen los resultados esperados

Tabla 4.1: Casos de prueba comunes

2. Autenticación	
ID	Condición a cumplir
2.1	No se permite el acceso a ninguna otra sección sin una autenticación correcta
2.2	La autenticación no resulta exitosa con una clave incorrecta
2.3	Cada usuario solo puede acceder a la parte de la aplicación que le corresponde a su rol
2.4	La autenticación para usuario de tipo docente se bloquea tras 5 intentos fallidos durante 5 minutos
2.5	Si las credenciales no son correctas no se informa de la existencia del usuario

2.6	Solo se proporciona un usuario a los alumnos si el expediente introducido está matriculado en la asignatura seleccionada
2.7	Solo se recuerda el usuario de un alumno si el expediente introducido está matriculado en la asignatura seleccionada
2.8	El usuario generado para un alumno es accesible inmediatamente después

Tabla 4.2: Casos de prueba para la autenticación

3. Gestión de asignaturas	
ID	Condición a cumplir
3.1	No se permiten crear asignaturas con el mismo código
3.2	El listado se agrupa correctamente por el curso y titulación correspondiente a cada asignatura
3.3	En la edición de la asignatura el código no se puede modificar
3.4	No se permite especificar un umbral fuera del rango entre 0 y 1 ambos incluidos
3.5	No se puede eliminar una asignatura con alumnos matriculados
3.6	En la ficha de la asignatura las pruebas asignadas se muestran agrupadas por convocatoria correctamente
3.7	El botón de cambio de visibilidad de la asignación funciona correctamente
3.8	Se puede establecer un umbral a la asignación solo con valores entre 0 y 1 ambos incluidos

Tabla 4.3: Casos de prueba para la gestión de asignaturas

4. Gestión de alumnos y matrículas	
ID	Condición a cumplir
4.1	El número de expediente y el DNI no se pueden repetir para más de 1 alumno
4.2	Un alumno puede ser matriculado en varias asignaturas
4.3	Un alumno no puede ser matriculado dos veces en una misma asignatura
4.4	Al matricular alumnos seleccionados en el listado de alumnos se matriculan correctamente solo y cada uno de los alumnos seleccionados en la asignatura elegida
4.5	Se permite editar cualquier dato del alumno

4.6	Un alumno no se puede eliminar si está matriculado en alguna asignatura o ha realizado alguna prueba
4.7	Si al importar un fichero de alumnos este tiene fallos o le faltan campos se notificará sin importar nada
4.8	Al importar alumnos los que ya existan se ignoraran mostrando el número de importados con respecto al total
4.9	Al importar un fichero de alumnos si se selecciona una asignatura se matricularán los que no estén matriculados y el resto se ignoran mostrando el número de importados y matriculados
4.10	No se puede eliminar la matrícula de un alumno que haya realizado alguna prueba

Tabla 4.4: Casos de prueba para la gestión de alumnos y matrículas

5. Gestión de pruebas, asignaciones	
ID	Condición a cumplir
5.1	No se puede editar el id de la prueba
5.2	La puntuación de las preguntas no puede ser negativa
5.3	No se pueden eliminar pruebas con asignaciones
5.4	No se puede cambiar el tipo de una prueba que ya este asignada
5.5	Si a una prueba se ha presentado algún alumno no se podrá editar ni el tipo ni el número de preguntas, solo corregir erratas
5.6	No se puede dar permiso de acceso extraordinario a un alumno si no está matriculado en una asignatura que la tenga asignada
5.7	No se puede asignar una prueba a una asignatura más de una vez
5.8	No se puede fijar una fecha de realización anterior a la actual al añadir o editar una asignación
5.9	Al asignar una prueba de tipo DDL a una asignatura se crean los usuarios de Oracle correspondientes
5.10	Al eliminar la asignación de una prueba de tipo DDL de una asignatura se borran los usuarios de Oracle correspondientes

Tabla 4.5: Casos de prueba para la gestión de pruebas y sus asignaciones

6. Gestión de entregas y corrección	
ID	Condición a cumplir
6.1	Los alumnos que se han presentado aparecen los primeros en la lista de entregas

6.2	Los alumnos que se han accedido varias veces aparecen destacados en el listado
6.3	La corrección automática de pruebas de tipo DML funciona correctamente
6.4	La corrección automática de pruebas DDL ejecuta el procedimiento específico correctamente
6.5	Solo se permite establecer la nota para cada pregunta de la entrega de un alumno con un valor entre 0 y su valor
6.6	En caso de estar una entrega corregida, la nota se corresponde con la suma de la nota de cada pregunta
6.7	Una entrega no se puede eliminar si está calificada
6.8	Para cada entrega se muestra información temporal y de identificación de red por cada vez que el alumno ha accedido a la prueba

Tabla 4.6: Casos de pruebas para la gestión de entregas y su corrección

7. Gestión de cuentas Oracle	
ID	Condición a cumplir
7.1	Al bloquear o desbloquear un usuario solo se aplica la acción para ese usuario
7.2	Al bloquear o desbloquear todos los usuarios de una asignatura la acción se realiza exclusivamente para dichos usuarios
7.3	Si un alumno ha sido matriculado después de la asignación de una prueba DDL, se puede crear el usuario para la prueba desde el listado
7.4	En el caso de las cuentas DDL se muestra la contraseña correcta para usuario en la base de datos

Tabla 4.7: Casos de prueba para la gestión de cuentas de Oracle

8. Panel y ficha del alumno	
ID	Condición a cumplir
8.1	Se muestran correctamente los datos del alumno autenticado y su asignatura correspondiente
8.2	En el listado de pruebas sólo aparecerán las pruebas marcadas como visible desde el panel del docente
8.3	Para cada asignatura aparece el estado correspondiente de la entrega y la corrección
8.4	Las pruebas aparecen correctamente agrupadas por convocatoria

Tabla 4.8: Casos de prueba para el panel y ficha del alumno

9. Realización y revisión de pruebas por parte del alumno	
ID	Condición a cumplir
9.1	El alumno solo podrá realizar pruebas si la fecha actual está entre la fecha de inicio y la fecha de fin de la asignación
9.2	El alumno podrá acceder todas las veces necesarias a la prueba mientras no la haya entregado
9.3	Si el alumno ha entregado una prueba no puede volver a realizarla
9.4	Solo se pueden revisar las pruebas asignadas que hayan sido corregidas y estén marcadas como visibles desde el panel del docente
9.5	El alumno solo puede realizar la prueba si la palabra de paso introducida coincide con la establecida en la asignación
9.6	Si se intenta entregar la prueba pasado el tiempo de finalización de la prueba, no se permite hacer la entrega mostrando una notificación sin perder las soluciones propuestas
9.7	En la revisión de pruebas DML se muestra el la respuesta y la solución para cada pregunta
9.8	En la revisión de pruebas DDL no se muestra respuesta ni solución
9.9	En la revisión de una prueba la calificación total se corresponde con la suma de las calificaciones para cada pregunta de la entrega

Tabla 4.9: Casos de prueba para la realización y revisión de pruebas por el alumno

10. Gestión de notas externas	
ID	Condición a cumplir
10.1	La prueba externa queda vinculada únicamente a la asignatura para la que se crea
10.2	Si al importar un fichero con notas hay algún error o faltan campos en el fichero, se notifica sin importar nada
10.3	Si al importar un fichero con notas alguno de los alumnos incluidos no se encuentra matriculado en la asignatura, se omiten importando el resto correctamente y muestra el número de importados
10.4	Se puede editar manualmente la nota de un alumno aunque se haya introducido otra previamente
10.5	Se puede eliminar la nota asignada a un alumno en concreto
10.6	La importación de ficheros con calificaciones funciona correctamente con todas sus opciones
10.7	Al importar un fichero con notas se sobrescriben las existentes para los alumnos incluidos en el fichero, sin afectar al resto

Tabla 4.10: Casos de prueba para la gestión de notas externas

11. Generación de calificaciones finales de una asignatura	
ID	Condición a cumplir
11.1	Sólo se pueden generar para las convocatorias en las que la asignatura tenga pruebas asignadas, notificando si no tiene
11.2	Las calificaciones generadas quedan guardadas y se muestran junto con la fecha de generación
11.3	Al volver a generar las calificaciones para una convocatoria que ya habían sido generadas, se eliminan las notas existentes y se actualiza la fecha de generación
11.4	Al editar manualmente la calificación final de un alumno se muestra en el informe la fecha de la última revisión correctamente
11.5	A la hora de generar las calificaciones finales se tienen en cuenta las notas tanto de las pruebas asignadas como de las pruebas externas de la asignatura
11.6	Los umbrales se aplican correctamente a la hora del cálculo de la nota, calificando al alumno con un 4.50 si no se superan
11.7	La calificación asignada manualmente prevalece sobre la calculada automáticamente
11.8	Si se ha suspendido a un alumno a causa del umbral teniendo la media aprobada, aparece un símbolo que lo indica junto a la calificación
11.9	Si se ha editado manualmente la calificación generada, se muestra un icono que lo indica junto a la calificación
11.10	El valor de las calificaciones para cada alumno es correcto
11.11	Sólo se tienen en cuenta las pruebas internas y externas asignadas a la convocatoria seleccionada

Tabla 4.11: Casos de prueba para la generación de calificaciones finales

12. Generación de informes	
ID	Condición a cumplir
12.1	Solo se permiten consultas SQL de tipo SELECT para el informe personalizado
12.2	En el informe personalizado, si la consulta introducida provoca un error de base de datos se muestra una alerta con el error correspondiente
12.3	Se pueden mostrar u ocultar las columnas del informe después de ser generado
12.4	La exportación a los distintos tipos funciona correctamente

12.5	La acción de imprimir lanza la herramienta de impresión del navegador
12.6	A la hora de exportar o imprimir informes sólo se tienen en cuenta las columnas marcadas como visible
12.7	Al seleccionar la asignatura para el informe de calificaciones por prueba solo aparecen las pruebas que la asignatura tiene asignadas
12.8	El informe de calificaciones por convocatoria solo estará disponible si las calificaciones finales para la asignatura y convocatoria elegidas han sido generadas previamente notificándolo en caso contrario

Tabla 4.12: Casos de prueba para la generación de informes

13.Edición del menú de alumnos desde el panel docente	
ID	Condición a cumplir
13.1	Se pueden incluir enlaces tanto a páginas internas de la aplicación como a sitios externos
13.2	La activación y desactivación para cada enlace funciona correctamente

Tabla 4.13: Casos de prueba para la configuración desde el panel de docente

4.8.2 Pruebas de carga

Para comprobar que la aplicación es capaz de soportar un mínimo de usuarios concurrentes sin que deje de estar operativa ni presente errores a los usuarios se realizan pruebas que simulen una carga y comportamientos sobre el servidor semejantes a situaciones extremas.

Para ello utilizamos Apache JMeter, una herramienta que nos permite crear un flujo de navegación (o grabarlo con posibilidad de editarlo) y reproducirlo con múltiples hilos concurrentes y un amplio abanico de configuraciones para simular escenarios reales.

Para este caso se ha generado un CSV con datos aleatorios de alumnos con la herramienta Mockaroo [7] y se ha importado y matriculado a una asignatura de prueba en la aplicación desde el panel docente de la propia aplicación. Generamos mediante consultas SQL los usuarios y claves para cada matrícula (activamos uno de los usuarios y copiamos la misma clave, ya que es un hash generado por la aplicación) y los exportamos a un fichero CSV para que JMeter utilice un usuario distinto por cada hilo y puedan realizar el flujo completo de la

entrega de la prueba, de manera que tengamos en cuenta el impacto de los accesos a la base de datos.

Con la ayuda del proxy que proporciona la herramienta se graba el flujo completo de la realización de una prueba y se realizan las modificaciones necesarias para automatizarlo quedando como en la siguiente imagen.

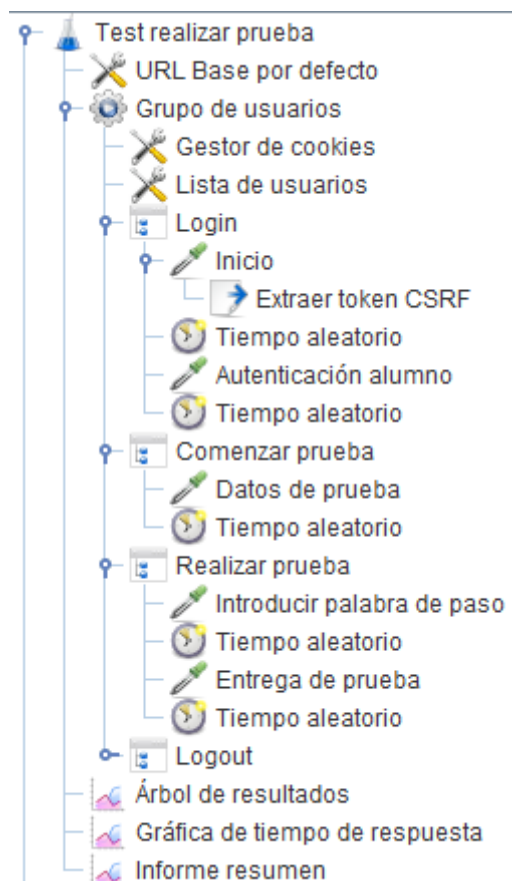


Ilustración 4.8: Flujo de prueba de carga en Apache JMeter

Entre cada petición introducimos un tiempo de espera aleatorio entre 1 y 10 segundos que le darán más realismo que si se hicieran todas las peticiones seguidas.

Ejecutamos la prueba configurando el grupo de hilos con 150 hilos que comienzan aleatoriamente en un rango de 60 segundos.

Cuando finaliza el test a los 2 minutos aproximadamente la herramienta nos proporciona los distintos informes que se han configurado. Comprobamos a simple vista en la gráfica que los tiempos de respuesta del servidor se han

mantenido en un rango normal durante toda la prueba, los cuales no superan los 600 milisegundos de tiempo de respuesta.

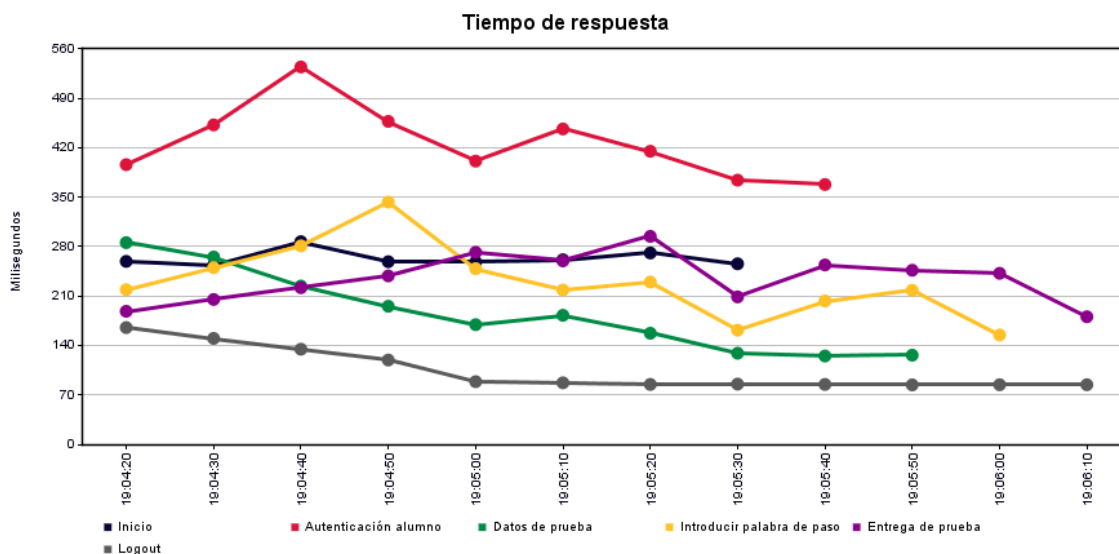


Ilustración 4.9: Gráfica de tiempos de respuesta de la prueba de carga con JMeter

Podemos ver los resultados con más detalle en las tablas resumen que incluyen datos estadísticos con respecto al tiempo de respuesta e información sobre errores y consumo de ancho de banda.

Se puede observar que los tiempos de respuesta son muy bajos estando la media en 231 milisegundos y el máximo en 880 milisegundos, contando con que la prueba se ha realizado desde fuera de la Universidad. Esto indica que el objetivo de hacer una aplicación ligera y rápida se ha conseguido.

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar
Inicio	150	265	243	747	44,25
Autenticación ...	150	424	260	880	107,67
Datos de prue...	150	160	44	415	66,97
Introducir pala...	150	210	130	671	95,73
Entrega de pru...	150	242	172	729	93,04
Logout	150	84	81	96	3,04
Total	900	231	44	880	129,94

Ilustración 4.10: Primera tabla resumen con datos estadísticos de la prueba de carga con JMeter

Etiqueta	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes
Inicio	0,00%	2,2/sec	11,61	0,27	5293,0
Autenticación ...	0,00%	2,2/sec	15,86	1,16	7510,0
Datos de prue...	0,00%	2,1/sec	15,08	0,39	7510,0
Introducir pala...	0,00%	1,9/sec	23,22	0,56	12488,4
Entrega de pru...	0,00%	1,9/sec	13,75	1,00	7591,0
Logout	0,00%	1,9/sec	10,26	0,77	5649,0
Total	0,00%	8,0/sec	60,17	2,80	7673,6

Ilustración 4.11: Segunda tabla resumen con datos estadísticos de la prueba de carga con JMeter

En la segunda tabla vemos que todas las peticiones han devuelto un código de respuesta correcto, lo que indica que se ha seguido el flujo correctamente sin errores.

Además, también se puede ver para cada petición realizada al servidor el detalle tanto de la petición como de la respuesta devuelta por el servidor, lo cual permite comprobar el correcto funcionamiento de la aplicación y que cada hilo se ha ejecutado con un usuario y una sesión distinta.

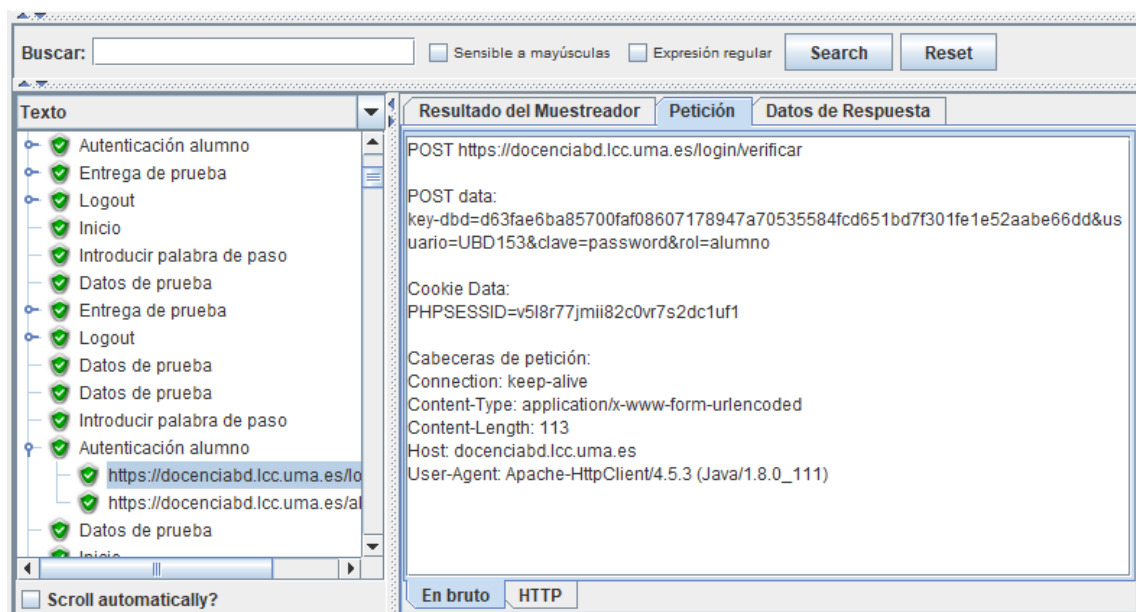



Ilustración 4.12: Árbol de peticiones en la prueba de carga con JMeter

Con las opciones que ofrece esta herramienta no sólo se consiguen hacer pruebas de carga, sino también pruebas funcionales con asertos e indicadores de varios tipos, permitiendo realizar una simulación completa.

Se concluye la prueba con un resultado satisfactorio y que queda reflejado en la vista de entregas del panel de docente de la aplicación, que muestra el listado de entregas de todos los alumnos para la prueba y la asignatura que hemos utilizado en este caso.


Docencia de Bases de Datos
 Departamento de lenguajes y ciencias de la computación

tfg_gamarro
 Profesor
[Salir](#)

▶ Resumen
 ▶ Asignaturas
 ▶ Alumnos
 ▶ Pruebas
 ▶ Notas Externas
 ▶ Cuentas
 ▶ Informes
 ▶ Configuración

Panel docente / Gestión de pruebas

Listado
 + Nueva
 Editar
 Asignaciones
 ✓ Corrección

« Ir a las asignaciones de la prueba
 Ir a la ficha de la asignatura »

Asignatura	TEST Bases de datos (B, 17/18)	Conv.	FEBRERO
Prueba	Prueba de consultas 1 (1)	Tipo	DML
Fecha inicio	16/09/2018 00:00	Fecha fin	31/10/2018 00:00
		Umbral	0,50

Entregas

Corrección automática

Mostrar 10 registros
 Buscar:

Apellidos	Nombre	Nota	
Agastina Nora	Hiland	-	Accesos Revisar
Aguirre Samuele	Erricker	-	Accesos Revisar
Akast Enos	Grey	-	Accesos Revisar
Akid Augustina	Dundendale	-	Accesos Revisar
Allawy Nelly	Pakenham	-	Accesos Revisar
Anetts Frannie	Burberow	-	Accesos Revisar
Augustus Andy	Rahl	-	Accesos Revisar
Backman Wendel	Parkman	-	Accesos Revisar
Badham Harlan	Abram	-	Accesos Revisar
Barter Freida	Giacopetti	-	Accesos Revisar

Mostrando registros del 1 al 10 de un total de 150 registros
 Anterior 1 2 3 4 5 ... 15 Siguiente

Ilustración 4.13: Listado de entregas resultantes de la prueba de carga

5 Conclusiones y trabajo futuro

La realización de este proyecto ha permitido adentrarse en el desarrollo web abarcando todos sus componentes, desde el nivel de sistemas o infraestructura al haber configurado y desplegado la máquina virtual que hace de servidor, hasta la programación de lado del cliente que interactúa directamente en el navegador con el usuario. Entre esos dos extremos se encuentra la aplicación que se ejecuta en el servidor y la base de datos a la que se conecta, que son el objeto y los principales protagonistas del trabajo.

Se ha profundizado en el desarrollo de aplicaciones con PHP y Javascript, que son de lo más utilizados en el desarrollo web, y en el patrón de diseño Modelo-Vista-Controlador aplicando la estructura utilizada en la mayoría de los frameworks para aplicaciones de este tipo.

El estudio de estas tecnologías junto con la realización de operaciones avanzadas con bases de datos Oracle, que es de la más reconocidas en el ámbito profesional, ha proporcionado abundantes y valiosos conocimientos que facilitan mucho la incorporación al mercado laboral.

Después de todo el trabajo se ha obtenido un resultado satisfactorio obteniendo una aplicación para un uso real y acorde a lo que se necesitaba, aportando una mejora al proceso de evaluación tanto para profesores como para alumnos.

Además de conocimientos técnicos y satisfacción, la realización del trabajo también ha aportado madurez y capacidad a la hora de organizar proyectos y redactar documentos extensos como este.

La aplicación está pensada para que se puedan añadir nuevos módulos o secciones fácilmente, por lo que puede servir de base para otros Trabajo de fin de Grado o los propios docentes pueden ampliarla de acorde a sus necesidades fácilmente. Algunas funciones que sería interesante añadir son:

- Creación de pruebas aleatorias para pruebas prácticas en base a un gran conjunto de preguntas
- Realización de pruebas tipo test
- Estadísticas de acierto por preguntas, tipos de pruebas, titulaciones...

6 Referencias

- [1] «GUMP for PHP»
<https://github.com/Wixel/GUMP>
- [2] «PhpSpreadsheet,»
<https://github.com/PHPOffice/PhpSpreadsheet>
- [3] «php-csrf»
<https://github.com/GramThanos/php-csrf/blob/master/php-csrf.php>
- [4] «Bootstrap»
<http://getbootstrap.com/>
- [5] «Oracle OCI8»
<http://php.net/manual/es/book.oci8.php>
- [6] «Using PHP with Oracle Database 11g»
https://www.oracle.com/webfolder/technetwork/tutorials/obe/db/oow10/php_db/php_db.htm
- [7] «Mockaroo»
<https://www.mockaroo.com/>
- [8] «Documentación de PHP»
<http://php.net/manual/es/>
- [9] C. J. a. A. Holloway, The Underground PHP and Oracle Manual, ORACLE, 2012.

7 Anexos

7.1 Manual de usuario

En este anexo se realiza un recorrido guiado por las funciones de la aplicación que componen el flujo principal de cada uno de los tipos de usuarios que la utilizan.

7.1.1 Manual para usuarios con rol Docente

Para este tipo de usuarios procedemos a explicar las funcionalidades que puede realizar en el orden correspondiente al proceso de evaluación normal de una asignatura.

Vamos a simular la evaluación para una convocatoria que comprende una prueba de tipo DML y las notas externas de un examen de tipo test realizado a través del campus virtual.

El docente deberá autenticarse con las credenciales proporcionadas por el administrador de la aplicación, accediendo directamente al listado de pruebas.

The screenshot displays the 'Docencia de Bases de Datos' application interface. The header includes the logo, the title 'Docencia de Bases de Datos', the subtitle 'Departamento de lenguajes y ciencias de la computación', and a user profile section with 'prueba Docente' and a 'Salir' button. The left sidebar contains a menu with 'Asignaturas', 'Alumnos', 'Pruebas', 'Notas Externas', 'Cuentas', 'Informes', and 'Configuración'. The main content area is titled 'Panel docente / Gestión de asignaturas' and features a toolbar with 'Listado', '+ Nueva', 'Editar', 'Ficha', and 'Calificaciones'. Below the toolbar, there is a 'Curso:' dropdown set to '18/19' and a 'Buscar:' input field. The main table lists assignments with columns for '#', 'Nombre', 'Grupo', 'Cuat.', and 'Pública'. The table is organized into sections: 'Ingeniería de caminos' (PBD), 'Ingeniería del software' (AAA, BDB), 'Ingeniería informática' (ABDA), and 'Métodos estadísticos' (FBD). Each row includes 'Editar', 'Ficha', and 'Calificaciones' buttons.

#	Nombre	Grupo	Cuat.	Pública	
Ingeniería de caminos					
PBD	Principios de las Bases de Datos	A	1	No	Editar Ficha Calificaciones
Ingeniería del software					
AAA	Administración Acciones Acid	A	1	Sí	Editar Ficha Calificaciones
BDB	Bases de Datos	B	1	Sí	Editar Ficha Calificaciones
Ingeniería informática					
ABDA	Administración de base de datos	A	2	Sí	Editar Ficha Calificaciones
Métodos estadísticos					
FBD	Fundamentos de Bases de Datos	B	2	No	Editar Ficha Calificaciones

Desde las pestañas superiores del panel principal, el usuario puede acceder a la creación de una nueva prueba mediante un formulario.

Una vez que se ha creado la prueba nos dirigimos a la pestaña “Añadir” de la sección “Alumnos”, donde podremos importar un fichero de alumnos con el formato que se indica al pulsar sobre la etiqueta “Columnas”.

Antes de pulsar el botón importar seleccionamos la asignatura que hemos creado para matricular a los alumnos directamente.

Llegados a este punto podemos ver la lista de usuarios de la asignatura utilizando los filtros del listado de la sección “Alumnos”, pudiendo acceder a la ficha de cada uno para editar sus datos personales e identificativos en la aplicación.

Apellidos	Nombre	Expediente	DNI	
Allwright Chancey	Geall	1234574	26739288P	Ver ficha Editar
Archbould Reed	Lory	1234578	48258086V	Ver ficha Editar
Bagniuik Bentley	Ashingden	1234573	49525061C	Ver ficha Editar
Baldung Ronna	Dye	1234586	27762852U	Ver ficha Editar

El siguiente paso es la creación de la prueba a realizar por los alumnos, lo cual se puede hacer desde la pestaña “Nueva” de la sección “Pruebas”.


The screenshot shows the 'Docencia de Bases de Datos' web application interface. The header includes the logo and name of the department, and a user profile with a 'Salir' button. The left sidebar contains navigation links for 'Asignaturas', 'Alumnos', 'Pruebas', 'Notas Externas', 'Cuentas', 'Informes', and 'Configuración'. The main content area is titled 'Panel docente / Gestión de pruebas' and features a 'Listado' tab and a '+ Nueva' button. Below this, the 'Información' section contains fields for 'Tipo' (set to 'DDL'), 'Nombre' ('Prueba para manual'), and 'Descripción' ('Esta prueba se ha creado para la realización del manual de usuario de esta aplicación'). The 'Preguntas' section shows a list of questions (1, 2, 3, 4) with 'Eliminar pregunta' and 'Añadir pregunta' buttons. The 'Puntos' field is set to '2.5' and the 'Esquema' field contains 'NombreAsignatura,NombreAlumno,Apellido1,Apellido2'. The 'Enunciado' field contains the text: 'OBTENGA EL NOMBRE DE CADA ASIGNATURA JUNTO CON EL NOMBRE Y APELLIDOS DEL ALUMNO MÁS JOVEN MATRICULADO EN ELLA. NO MUESTRE INFORMACIÓN DUPLICADA'. The 'Solución' field contains a SQL query: 'select distinct a.nombre "NombreAsignatura", al.nombre "NombreAlumno", apellido1, apellido2 from asignaturas a, matricular m, alumnos al where a.CODIGO=m.ASIGNATURA and m.ALUMNO=al.DNI and fecha_nacimiento = (select max(fecha_nacimiento) from matricular, alumnos where asignatura=a.CODIGO and alumno=dni);'.



La vista de creación de la prueba permite cambiar dinámicamente el número de preguntas e incluye dos campos adicionales al seleccionar el tipo DDL, que son el prefijo para los usuarios destinados a realizar dicha prueba y el nombre del procedimiento de autocorrección.

Al contenido del enunciado se le pueden aplicar estilos, así como añadirle imágenes o enlaces mediante la barra de herramientas que se incluye en el campo de texto.

Después de crear la prueba la buscamos en el listado y pulsamos su botón de “Asignaciones” para ir donde podremos asignar la prueba a nuestra asignatura.

Para ello pulsamos el botón “Añadir” del listado de asignaciones de la prueba, que desplegará una modal donde elegimos la asignatura e introducimos las fechas de inicio y fin y la palabra de paso, así como la convocatoria para la que se tiene en cuenta.


Docencia de Bases de Datos
Departamento de lenguajes y ciencias de la computación

prueba 
Docente 
Salir 

- Asignaturas
- Alumnos
- Pruebas
- Notas Externas
- Cuentas
- Informes
- Configuración

Panel docente / Gestión de pruebas

Prueba asignada correctamente

Listado
+ Nueva
Editar
Asignaciones
Corrección

Prueba Prueba para manual (26)
Tipo DML

Descripción Esta prueba se ha creado para la realización del manual de usuario de esta aplicación

Asignaciones a asignaturas

Curso: 18/19
Convocatoria: Todas

Asignatura	Inicio	Fin	
Manual Bases de Datos (C)	23/09/2018 17:40	30/09/2018 21:00	Visible Editar Corrección


Asignaciones extra a alumnos




No hay permisos extras concedidos

Para que los alumnos puedan ver la prueba habrá que activar el check animado que aparece junto a la asignación con el nombre de “Visible”.

Ahora que la prueba está preparada, en el momento de comenzar el examen el profesor tendrá que activar las cuentas de Oracle de los alumnos para que puedan ejecutar las consultas en la base de datos.

Esto se hace desde la sección “Cuentas” del menú, la cual nos mostrará un panel con pestañas para cada tipo de usuarios. En la pestaña de usuarios DML podremos obtener el listado de los usuarios para los alumnos de la asignatura que seleccionemos.


Docencia de Bases de Datos
Departamento de lenguajes y ciencias de la computación

prueba 
Docente 
Salir 

- Asignaturas
- Alumnos
- Pruebas
- Notas Externas
- Cuentas
- Informes
- Configuración

Panel docente / Gestión de cuentas de Oracle

★ Prácticas / DML
★ DDL

Manual Bases de Datos (C, 18/19)

Mostrar 10 registros
Bloquear todas
Desbloquear todas
Buscar:

Expediente	Apellidos	Nombre	Usuario	
1234574	Allwright Chancey	Geall	UBD626	Bloquear Desbloquear
1234578	Archbould Reed	Lory	UBD630	Bloquear Desbloquear
1234573	Bagniuk Bentley	Ashingden	UBD625	Bloquear Desbloquear
1234586	Baldung Ronna	Dye	UBD638	Bloquear Desbloquear
1234601	Biasotti Damita	Creeboe	UBD653	Bloquear Desbloquear

Junto al listado de usuarios hay disponibles botones para bloquear y desbloquear las cuentas individualmente o para todos los alumnos de la asignatura. La acción de estos botones será inmediata.

Una vez la prueba ha sido realizada, el docente puede revisar la lista de entregas a la que puede acceder a través de la ficha de la asignatura o desde el listado de asignaciones para la prueba con el botón “Asignaciones” que las acompaña en ambos casos.

En este listado se puede ver la fecha y dirección IP de cada acceso que ha realizado el alumno a dicha prueba antes de entregarla pulsando sobre la etiqueta “Accesos” de cada alumno, la cual aparecerá en rojo si han sido varios.

Docencia de Bases de Datos
Departamento de lenguajes y ciencias de la computación

prueba Docente Salir

Panel docente / Gestión de pruebas

Listado Nueva Editar Asignaciones Corrección

« Ir a las asignaciones de la prueba Ir a la ficha de la asignatura »

Asignatura	Manual Bases de Datos (C, 18/19)	Conv.	Septiembre
Prueba	Prueba para manual (26)	Tipo	DML
Fecha inicio	23/09/2018 17:40	Fecha fin	30/09/2018 21:00
Umbral	0,50		

Entregas Corrección automática

Mostrar 10 registros Buscar:

Apellidos	Nombre	Nota		
Allwright Chancey	Geall	5,01	Accesos	Revisar
Archbould Reed	Lory	5,48	Accesos	Revisar
Bagniuk Bentley	Ashingden	2,98	Accesos	Revisar
Baldung Ronna	Dye	4,38	Accesos	Revisar
Biasotti Damita	Creeboe	5,20	Accesos	Revisar
Bird Lexie	Sorley	3,12	Accesos	Revisar
Bohlens Fayina	Pury	4,84	Accesos	Revisar
Christopherson Waverly	Seawell	4,62	Accesos	Revisar

Para lanzar la corrección automática de las entregas de la asignación pulsamos el botón “Corrección automática”, que ejecutará el procedimiento necesario.

Además, es posible revisar la entrega de un alumno desde el botón “Revisar” que desplegará una modal con las respuestas para cada pregunta, información de la prueba y la acción de cambiar la calificación de cada pregunta manualmente.

El otro componente de la evaluación de la asignatura son las notas realizadas a través de medios externos a la aplicación, las cuales pueden ser registradas en la aplicación junto con sus notas desde la sección “Notas Externas”.

Desde la pestaña “Nueva” creamos una prueba externa para nuestra asignatura estableciendo la convocatoria en la que se tiene en cuenta junto con el resto de los datos que la identifican.

Docencia de Bases de Datos
Departamento de lenguajes y ciencias de la computación

prueba
Docente
Salir

Panel docente / Gestión de notas externas

Pruebas + Nueva Editar Notas

Asignatura: Manual Bases de Datos (C, 18/19)

Nombre: Prueba externa para manual

Descripción: Prueba externa de tipo test para el manual de usuario de la aplicación

Puntos: 10 Umbral: 0.5

Convocatoria: Septiembre Tipo: TET

Fecha: 12/09/2018 18:00

Guardar Limpiar Cancelar

Importamos las notas externas a través de un fichero desde la pestaña “Notas” de la sección “Notas externas”, donde también podemos editarlas manualmente.

Panel docente / Gestión de notas externas

50 notas importadas correctamente

Pruebas + Nueva Editar Notas

« Editar los datos de la prueba Ir a la ficha de la asignatura »

Prueba: Prueba externa para manual (26) Tipo: TET

Asignatura: Manual Bases de Datos (C, 18/19) Fecha: 12/09/2018 18:00

Convocatoria: Septiembre Puntuación: 10

50 alumnos con calificación de 50

Mostrar 10 registros Buscar:

Expediente	Apellidos	Nombre	Nota
1234574	Allwright Chancey	Geall	3.13
1234578	Archbould Reed	Lory	.35
1234573	Bagniuk Bentley	Ashingden	3.42
1234586	Baldung Ronna	Dye	
1234601	Biasotti Damita	Creeboe	
1234594	Bird Lexie	Sorley	9.85

Editar nota

9.85

En la pestaña “Ficha” de la sección “Asignaturas” podemos ver un resumen de las pruebas asignadas a la asignatura elegida para cada convocatoria. Para cada prueba se muestra su valor y se permite cambiar el umbral mínimo a obtener en esa prueba para superar la asignatura.

El umbral permite que la aplicación suspenda automáticamente al alumno al calcular las notas finales, y en la ficha de la asignatura también se puede editar el umbral de la asignatura que se utilizará por defecto si la asignación no tiene uno especificado.

Docencia de Bases de Datos
Departamento de lenguajes y ciencias de la computación

prueba Docente Salir

Panel docente / Gestión de asignaturas

Asignaturas Alumnos Pruebas Notas Externas Cuentas Informes Configuración

Listado Nueva Editar **Ficha** Calificaciones

« Editar los datos de la asignatura Ir a las calificaciones de la asignatura »

Asignatura	Manual Bases de Datos	Grupo	C
Titulación	Ingeniería Informática	Curso	18/19
Umbral	0.5	Alumnos	50

Pruebas asignadas por convocatoria

Febrero / Junio **Septiembre** Diciembre Sin convocatoria

#	Nombre	Tipo	Puntos	Umbral	Visible	Corrección
26	Prueba para manual	DML	10	0.50	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
26	Prueba externa para manual	TET	10	0.50	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

La única acción que nos falta para finalizar el proceso de evaluación de la asignatura es calcular las notas finales para la convocatoria en cuestión y generar un informe con las calificaciones.

Para ello vamos a la pestaña “Calificaciones” de la sección “Asignaturas” y seleccionamos la convocatoria correspondiente. Nos aparecerá un botón para generar las calificaciones si aún no se han generado, el cuál será necesario volver a ejecutar si se han realizado cambios en las calificaciones parciales después de su generación.

Se mostrará un listado con las calificaciones para cada alumno, que pueden ir acompañadas de un indicador rojo en forma de guion si se ha suspendido a causa de no superar algún umbral resultando su media aprobada.

Asignaturas
Alumnos
Pruebas
Notas Externas
Cuentas
Informes
Configuración

Panel docente / Gestión de asignaturas

Listado
Nueva
Editar
Ficha
Calificaciones

« Ir a la ficha de la asignatura

Asignatura
Manual Bases de Datos (Grupo C, 18/19)

Convocatoria
Septiembre

Convocatoria
Septiembre
Fecha generación
23/09/018 18:10
Última revisión
23/09/018 18:11
Volver a generar
Borraré las notas actuales

Mostrar 10 registros
Buscar:

DNI	Alumno	Nota	
26739288P	Allwright Chancey, Geall	5,00 ↑	Revisar
48258086V	Archbould Reed, Lory	2,92	Revisar
49525061C	Bagniuik Bentley, Ashingden	3,20	Revisar
27762852U	Baldung Ronna, Dye	2,77	Revisar
32004792Q	Biasotti Damita, Creeboe	7,40	Revisar
81847083I	Bird Lexie, Sorley	4,50 -	Revisar
14794398L	Bohlens Fayina, Pury	4,50 -	Revisar

Desde la misma vista podemos revisar y cambiar manualmente la calificación final de un alumno pulsando el botón “Revisar” que acompaña a su calificación, lo cuál desplegará una ventana modal con el resumen de calificaciones para las pruebas parciales de la convocatoria y las finales del resto de convocatorias.

La calificación establecida manualmente prevalece sobre la generada automáticamente, indicándose en el listado con un icono con forma de flecha junto a la calificación.

Se puede exportar el listado de notas parciales y finales desde la sección “Informes”, donde se permite mostrar y ocultar columnas antes de exportar y distintos formatos como PDF, CSV y Excel, además de la impresión directa.

Asignaturas
Alumnos
Pruebas
Notas Externas
Cuentas
Informes
Configuración

Panel docente / Informes

Predefinidos
Personalizado

Calificaciones por prueba
Calificaciones por convocatoria

Manual Bases de Datos (C, 18/19)
Septiembre

Asignatura
Manual Bases de Datos (Grupo C, 18/19)
Convocatoria
Septiembre

Columnas visibles
PDF
CSV
Excel
Imprimir
Buscar:

DNI	Expediente	Apellidos	Nombre	Nota
26739288P	1234574	Allwright Chancey	Geall	5,00
48258086V	1234578	Archbould Reed	Lory	2,92

7.1.2 Manual para usuarios con rol Alumno

La funcionalidad del alumno es más reducida que la del docente y se limita a la realización y revisión de las pruebas asignadas a su asignatura.

Lo primero que el usuario necesita para acceder a la aplicación es solicitar su usuario, lo cual puede hacer desde el enlace disponible en la pantalla de autenticación que lo llevará al siguiente formulario.

Docencia de Bases de Datos
Departamento de lenguajes y ciencias de la computación

Solicitar usuario

Formulario con los siguientes campos:

- Usuario: 1234574
- Asignatura: Manual Bases de Datos (C, 18/19)
- Contraseña:
- Confirmación de contraseña:

Botón: Solicitar

[Volver al login](#)

Una vez ingresado su número de expediente y la contraseña deseada se proporcionará por pantalla el usuario correspondiente para la asignatura seleccionada siempre y cuando estuviese matriculado previamente.

Al autenticarse con el usuario obtenido accederá directamente a la lista de pruebas disponibles para realizar.

Docencia de Bases de Datos
Departamento de lenguajes y ciencias de la computación

Geall Allwright Chancey
Manual Bases de Datos
[Salir](#)

Panel alumno


- Panel alumno
- Campus Virtual
- Manual SQL
- Configuración BD

Ficha Realizar prueba Revisar prueba

Pruebas disponibles

Nombre	Tipo	Conv	Fecha inicio	Fecha fin	
Prueba para manual	DML	2	23/09/2018 17:40	30/09/2018 21:00	Realizar

Tras pulsar en realizar la prueba elegida es necesario introducir la palabra de paso para la prueba que el profesor a establecido, que normalmente será proporcionada al comienzo del examen.


Docencia de Bases de Datos
Departamento de lenguajes y ciencias de la computación

Geall Allwright Chancey
Manual Bases de Datos
Salir

- Panel alumno
- Campus Virtual
- Manual SQL
- Configuración BD

Ficha Realizar prueba Revisar prueba

Prueba	Prueba para manual	Tipo	DML
Descripción	Esta prueba se ha creado para la realización del manual de usuario de esta aplicación		
Fecha inicio	23/09/2018 17:40	Fecha fin	30/09/2018 21:00


Comenzar prueba

Palabra de paso

Ordenador

Comenzar Cancelar

El alumno accederá a un formulario con datos informativos de la prueba y las credenciales del usuario a utilizar en la base de datos.


Docencia de Bases de Datos
Departamento de lenguajes y ciencias de la computación

Geall Allwright Chancey
Manual Bases de Datos
Salir

- Panel alumno
- Campus Virtual
- Manual SQL
- Configuración BD

Ficha Realizar prueba Revisar prueba

Prueba	Prueba para manual	Tipo	DML
Descripción	Esta prueba se ha creado para la realización del manual de usuario de esta aplicación		
Fecha inicio	23/09/2018 17:40	Fecha fin	30/09/2018 21:00

Para realizar la prueba utilice su cuenta de Oracle de la asignatura (UBD626) y su contraseña

Pregunta 1
OBTenga EL NOMBRE DE CADA ASIGNATURA JUNTO CON EL NOMBRE Y APELLIDOS DEL ALUMNO MÁS JOVEN MATRICULADO EN ELLA. NO MUESTRE INFORMACIÓN DUPLICADA

Resposta...

Esquema NombreAsignatura,NombreAlumno,Apellido1,Apellido2
Puntuación 2,50

Pregunta 2
OBTenga NOMBRE Y NÚMERO DE CREDITOS DE LAS ASIGNATURAS QUE NO SON IMPARTIDAS POR PROFESORES CON MENOS DE 25 AÑOS DE EXPERIENCIA

Resposta...

Sobre cada pregunta se muestra su enunciado junto con la puntuación y el esquema que se ha de proporcionar la solución (el cual no siempre se indica), permitiendo introducir la solución para el caso de las pruebas de tipo DML.


Para efectuar la entrega de la prueba se pulsa el botón Guardar al final del formulario, lo que llevará al usuario de nuevo al listado de pruebas a realizar con la notificación de entrega correspondiente.

Mediante las pestañas superiores del panel central, el alumno puede ir a su ficha donde podrá ver sus datos personales y de la asignatura junto con la lista de pruebas asignadas para cada convocatoria.

The screenshot shows a web application interface for 'Docencia de Bases de Datos' (Database Teaching). The header includes the department name and a user profile for 'Geall Allwright Chancey'. A sidebar on the left contains navigation links: 'Panel alumno', 'Campus Virtual', 'Manual SQL', and 'Configuración BD'. The main content area has three tabs: 'Ficha' (selected), 'Realizar prueba', and 'Revisar prueba'. Under the 'Ficha' tab, there are two sections: 'Datos personales' and 'Datos de la asignatura'. The 'Datos personales' section displays fields for 'Expediente' (1234574), 'DNI' (26739288P), 'Nombre' (Geall), and 'Apellidos' (Allwright Chancey). The 'Datos de la asignatura' section displays 'Nombre' (Manual Bases de Datos), 'Grupo' (C), 'Curso' (18/19), 'Titulación' (Ingeniería Informática), and 'Cuatrimestre' (Primero). Below these is a section for 'Pruebas asignadas' with a sub-header for 'Febrero / Junio', 'Septiembre' (selected), 'Diciembre', and 'Sin convocatoria'. A table lists the assigned tests with columns for 'Nombre', 'Tipo', 'Entrega', and 'Calificación'. The table shows one entry: 'Prueba para manual' (DML) with a delivery date of '23/09/2018 17:58' and a grade of '5,01'. A 'Revisar' button is visible next to the grade.

Nombre	Tipo	Entrega	Calificación
Prueba para manual	DML	23/09/2018 17:58	5,01

Para cada prueba se muestra el estado de entrega y la calificación en caso de haber sido corregida. En este último caso también se podrá ver la revisión de la prueba mediante el botón Realizar a la derecha de su calificación.



Docencia de Bases de Datos
Departamento de lenguajes y ciencias de la computación

Geall Allwright Chancey
Manual Bases de Datos

[Salir](#)

- Panel alumno
- Campus Virtual
- Manual SQL
- Configuración BD

FichaRealizar pruebaRevisar prueba

Prueba	Prueba para manual	Tipo	DML
Descripción	Esta prueba se ha creado para la realización del manual de usuario de esta aplicación	Conv.	Septiembre
23/09/2018 17:46	Puntos	Fecha entrega 5,01 / 10,00	

Pregunta 1

Valor	2,50	Nota	0,94
Comentario	Sin comentarios		

Enunciado

OBTenga EL NOMBRE DE CADA ASIGNATURA JUNTO CON EL NOMBRE Y APELLIDOS DEL ALUMNO MÁS JOVEN MATRICULADO EN ELLA. NO MUESTRE INFORMACIÓN DUPLICADA

Respuesta

```
SELECT select_list
FROM table1 t_ alias1
WHERE expr operator
(SELECT column_list
```

Solución

```
select distinct a.nombre "NombreAsignatura", al.nombre "NombreAlumno", apellido1, apellido2
from asignaturas a, matricular m, alumnos al where a.CODIGO=m.ASIGNATURA and m.ALUMNO=al.DNI
and fecha_nacimiento = (select max(fecha_nacimiento) from matricular, alumnos
where asignatura=a.CODIGO and alumno=al.dni);
```

En la vista de revisión el alumno puede comparar su respuesta con la solución correcta de cada pregunta además de ver la calificación parcial de cada una.

En caso de las pruebas de tipo DDL sólo se muestra el enunciado y la calificación para cada pregunta.

7.2 Manual de instalación

Para la instalación de la aplicación el proceso es similar al de cualquier aplicación de este tipo con algunas particularidades en la base de datos y las configuraciones propias a la estructura de este proyecto.

Los requisitos mínimos necesarios son:

- Servidor web con PHP 5.6 o superior y la extensión **oci8** para conexiones y operaciones a bases de datos Oracle desde la aplicación.
- Una base de datos Oracle 11g o superior y, si ésta se encuentra en una máquina distinta al servidor web, la máquina del servidor tendrá que tener **Oracle Instant Client** instalado, que es el conjunto de librerías y utilidades

necesarias para establecer conexiones con bases de datos remotas Oracle.

Será necesario crear el esquema en la base de datos y subir los ficheros de la aplicación a un servidor web, además de configurar ambas partes para que se conecten entre sí.

Empezamos con la base de datos partiendo de que tenemos un usuario creado con los permisos básicos necesarios que son RESOURCE y CONNECT.

Importamos el esquema de la base de datos que incluye las tablas con sus relaciones y restricciones. El fichero es **esquema.sql** de la carpeta SQL.

Una vez tenemos el esquema importamos la parte de PL/SQL, que incluye los disparadores, procedimientos y funciones, incluidos en el fichero **pl.sql** de la carpeta SQL. Este contiene dos funciones que necesitan permisos especiales, VALIDA_USUARIO y USUARIO_ROL. Los permisos que hay que otorgar desde un usuario administrador de la base de datos son:

```
grant select on sys.user$ to USUARIO;  
grant select on sys.sysauth$ to USUARIO;  
grant execute on sys.DBMS_CRYPT to USUARIO;
```

Procedemos a subir los ficheros al servidor web, lo cual podemos hacer mediante FTP, SSH o copiándolos directamente si se tiene acceso físico.

La aplicación se encuentra dentro del directorio Código fuente, donde encontraremos dos directorios:

- **publico** – El directorio principal del servidor web tiene que apuntar a este directorio. En él se incluyen los ficheros accesibles al público.
- **aplicacion** – Contiene el código fuente de la aplicación y se recomienda que esté fuera del directorio publico por temas de seguridad. En este caso la ubicamos al mismo nivel que la carpeta publico quedando protegida ante una configuración incorrecta que permitiese servir este tipo de ficheros.

Establecemos la configuración de base de datos editando el fichero **aplicación/Configuracion/configDB.ini**, que contendrá variables para el

usuario, clave y cadena de conexión para cada conexión utilizada en la aplicación, en este caso sólo una.

```
[default]
usuario = "tfg"
password = "tfg"
conexion = "localhost/XE:POOLED"
```

Por último, establecemos la URL base desde la que se accede a la aplicación editando el fichero **aplicacion/Configuracion/config.ini** de la siguiente forma:

```
; URL base desde la que se accede a la aplicación
url = "https://docenciabd.lcc.uma.es/"
```

Ya debería estar la aplicación operativa, y para empezar a utilizarla sólo es necesario crear un usuario en la base de datos con el rol **ROL_DOCENTE** y usarlo para autenticarse como docente.

7.3 Manual para añadir nuevas funcionalidades

Suponiendo que se ha leído la memoria de este proyecto completa y se está familiarizado con la estructura seguida para el desarrollo de la aplicación y los componentes que la componen, haremos una breve explicación de cómo añadir nuevas funcionalidades a la aplicación.

En una aplicación web las posibles acciones se realizan cada una mediante una petición a una URL concreta, que puede devolver datos o simplemente realizar alguna operación.

Cada una de estas acciones en nuestra aplicación se corresponde con un método de una clase que extiende de Controlador, por lo que lo primero que habrá que hacer será crear un controlador o elegir uno existente, que dependerá de cómo queramos organizar el código y la URL que queremos asignarle.

La URL resultante para la acción se construirá de modo que los primeros segmentos corresponderán a la jerarquía de directorios que cuelga de Controladores donde se encuentra el controlador, a los que le siguen el nombre del controlador, el nombre del método y los parámetros de entrada. Por ejemplo:

```
https://docenciabd.lcc.uma.es/docente/asignaturas/editar/BDB
```

Se corresponde con el método “editar” del controlador “Asignaturas” que se encuentra en el directorio “docente” dentro de la carpeta Controladores, pasando como parámetro “BDB” que identifica a la entidad a editar.

Como ejemplo creamos un controlador con un método que obtenga un listado de asignaturas que no tienen alumnos matriculado. Creamos el fichero **aplicación/Controladores/Ejemplo.php** con el siguiente contenido:

```
<?php
namespace Controladores;

use Nucleo\Controlador;
use Modelos\Ejemplo as ModeloEjemplo;

class Ejemplo extends Controlador {

    public function index ($cuatrimestre = null) {

        $this->checkDocente() {

            $modelo = new ModeloEjemplo();

            $lista = $modelo->getAsignaturasSinMat($cuatrimestre);

            $datosVista = array(
                'cuatrimestre' => $cuatrimestre,
                'lista' => $lista
            );
            $this->vista->mostrar('ejemplo', $datosVista);

        }
    }
}
```

El método *index* es el método por defecto que se carga cuando no se especifica un método en la URI. Para el ejemplo como solo vamos a tener un método lo llamaremos de esa forma.

El método comprueba si el usuario está autenticado como docente mediante las funciones heredadas y obtiene los datos de la base de datos que se quieren procesar a través de la función de un modelo previamente instanciado.

Una vez obtenidos los datos, los muestra a través de una vista que se renderiza con las funciones que ofrece la clase *Nucleo\Vista*, de la cual tenemos una instancia inyectada.

Es importante observar que el método cuenta con un parámetro opcional que se corresponde con el número de cuatrimestre, permitiendo filtrar los resultados por este criterio en caso de que se especifique.

Creamos el modelo utilizado en el controlador correspondiente al fichero **apliacion/Modelos/Ejemplo.php** con el siguiente código:

```
<?php
namespace Modelos;

use Nucleo\Modelo,
    Nucleo\BD;

class Ejemplo extends Modelo {

    public function getAsignaturasSinMat ($cuatrimestre) {

        $conexion = BD::getConexion();

        $consulta = "SELECT NOMBRE, CURSO, GRUPO, TITULACION FROM
ASIGNATURAS
        WHERE CODIGO NOT IN (SELECT ASIGNATURA FROM MATRICULAS)";
        $parametros = array();

        if (is_numeric($cuatrimestre)) {
            $consulta .= " AND CUATRIMESTRE = :cuat";
            $parametros['cuat'] = $cuatrimestre;
        }
        $resultado = $conexion->queryResult($consulta, $parametros);

        return $resultado;
    }

}
```

El modelo se limita a realizar la consulta necesaria a la base de datos utilizando la clase *Nucleo\BD* para obtener los datos y devolverlos al controlador.

La consulta utiliza comodines para referenciar a los datos de entrada, que se pasan a la función de consulta en formato de array con el identificador de cada uno como clave.

Para tener todos los elementos solo nos queda la vista, que se guarda en el fichero **aplicación/vistas/ejemplo.php**.

```
<h2>Asignaturas sin alumnos matriculados</h2>

<?php if (!empty($cuatrimestre)) { ?>
<p>A continuación se muestran las asignaturas que no tienen alumnos
matriculados para el cuatrimestre <?= $cuatrimestre ?></p>
```

```

<?php } else { ?>
<p>A continuación se muestran las asignaturas que no tienen alumnos
matriculados para todos los cuatrimestres</p>
<?php } ?>

<p>
    Filtrar por cuatrimestre:
    <a href="/ejemplo">Ninguno</a> -
    <a href="/ejemplo/index/1">Primero</a> -
    <a href="/ejemplo/index/2">Segundo</a>
</p>
<hr>
<?php
if (!empty($lista)) {
    echo '<ul>';

    foreach ($lista as $asignatura) {
        echo '<li>'
            . $asignatura['NOMBRE']
            . '(Grupo ' . $asignatura['GRUPO']
            . ' de ' . $asignatura['CURSO'] . ')</li>';
    }

    echo '</ul>';
} else {
    echo '<p>No hay asignaturas sin alumnos matriculados</p>';
}

```

Vemos como en la vista se utilizan los valores del array de parámetros como variables y se pueden crear vistas dinámicas fácilmente gracias a la capacidad que tiene PHP de embeberse en el código HTML.

Ahora podemos acceder desde la URL dominio/ siempre que seamos un docente.



Docencia de Bases de Datos
Departamento de lenguajes y ciencias de la computación

prueba 
 Docente 
 Salir 

▶ Asignaturas
 ▶ Alumnos
 ▶ Pruebas
 ▶ Notas Externas
 ▶ Cuentas
 ▶ Informes
 ▶ Configuración

Asignaturas sin alumnos matriculados

A continuación se muestran las asignaturas que no tienen alumnos matriculados para todos los cuatrimestres

Filtrar por cuatrimestre: [Ninguno](#) - [Primero](#) - [Segundo](#)

- Fundamentos de Bases de Datos(Grupo B de 18/19)
- Principios de las Bases de Datos(Grupo A de 18/19)

Ilustración 7.1: Resultado para el ejemplo de añadir nuevas funcionalidades

7.4 Manual de configuración del servidor

En este anexo se detalla el procedimiento de creación y configuración de la máquina virtual que hará de servidor de la aplicación.

El hipervisor donde se alojará la máquina es VMware ESXi 6, por lo que la máquina virtual se ha creado con la herramienta VMWare Workstation con la opción de compatibilidad de hardware Workstation 10.0, que permite usarla en las versiones 5.5 y 6.0 de ESXi.

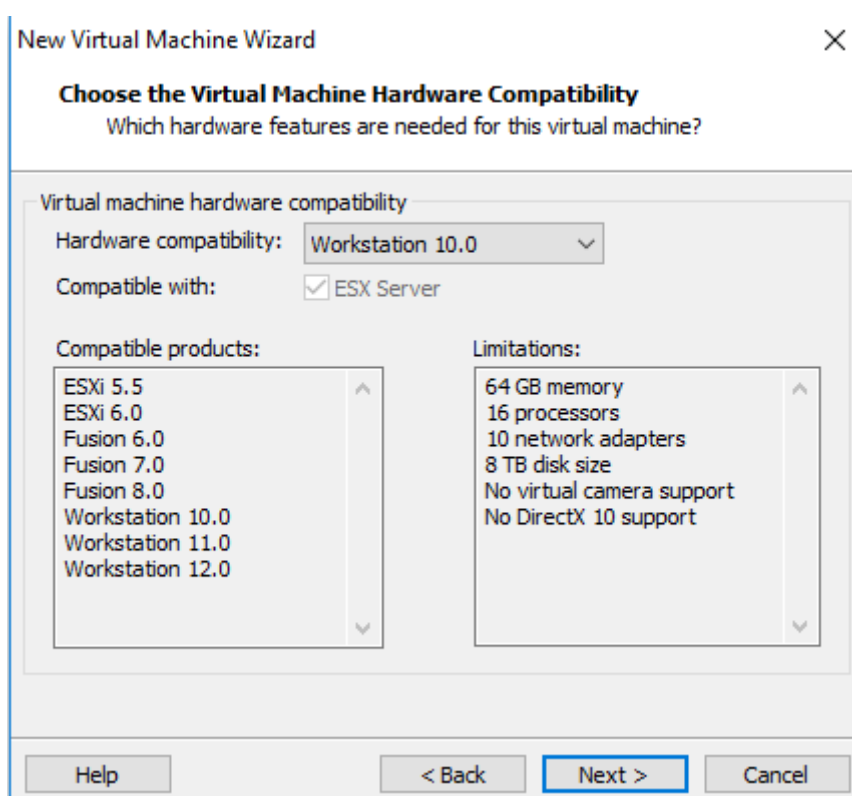


Ilustración 7.2: Configuración de compatibilidad para la máquina virtual

Descargamos la ISO de centos 7 minimal desde el sitio oficial y la seleccionamos como medio extraíble de nuestra máquina virtual, tras lo que procedemos a arrancarla.

Hacemos la instalación recomendada y cuando terminemos el asistente entraremos en el modo consola, el único que tiene esta versión minimal ya que no necesitamos interfaz gráfica para nada.

Lo primero que hacemos es activar la red y actualizar los repositorios. Lo primero lo hacemos con el comando **nmtui**, el cual nos permite activar la interfaz de red sencillamente a través de una interfaz de texto de usuario. Salimos y comprobamos que hemos obtenido una IP válida con **ip addr**.

```
$ nmtui
$ ip addr
```

Actualizamos los repositorios del sistema operativo:

```
$ yum update
```

Instalamos **nano** y **wget**, el primero para editar ficheros de texto de forma rápida y el segundo para descargar ficheros de servidores web desde consola.

```
$ yum install nano wget
```

Para obtener el mayor rendimiento de la máquina sobre hipervisores VMware instalamos las VMware Tools.

```
$ yum install open-vm-tools
$ shutdown -r
```

Instalamos **firewalld** y lo configuramos para permitir acceso a la máquina sólo por http y https, además de ssh que se permite por defecto.

```
$ yum install firewalld
$ firewall-cmd --permanent --zone=public --add-service=http
$ firewall-cmd --permanent --zone=public --add-service=https
$ firewall-cmd --reload
```

Instalamos y activamos el servidor web, en este caso Apache, que en CentOS se llama **httpd**.

```
$ yum install httpd
$ systemctl enable httpd
```

Descargamos los paquetes rpm (compatibles con CentOS) de **Oracle Instant Client**, lo cual lo haremos desde nuestro equipo y los copiaremos por SSH al servidor mediante la herramienta Bitvise SSH Client. Una vez los tenemos en la máquina virtual, los instalamos:

```
$ rpm -i oracle-instantclient12.2-basic-12.2.0.1.0-1.x86_64.rpm
$ rpm -I oracle-instantclient12.2-devel-12.2.0.1.0-1.x86_64.rpm
$ rpm -i oracle-instantclient12.2-sqlplus-12.2.0.1.0-1.x86_64.rpm
```

Para la correcta integración con PHP, necesitamos crear variables de entorno con los directorios principales de la instalación de Oracle Instant Client. Para ello creamos el fichero `/etc/profile.d/client.sh` con el siguiente contenido:

```
export ORACLE_HOME=/lib/oracle/12.2/client64
export PATH=$PATH:$ORACLE_HOME/bin
export LD_LIBRARY_PATH=$ORACLE_HOME/lib
```

Procedemos a instalar PHP, para lo que primero configuramos los repositorios donde se encuentran la versión deseada para luego proceder a la instalación de los paquetes necesarios:

```
$ rpm -Uvh https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
$ rpm -Uvh https://mirror.webtatic.com/yum/el7/webtatic-release.rpm
$ yum install -y php56w php56w-opcache php56w-xml php56w-mcrypt
php56w-devel php56w-intl php56w-mbstring
```

Para conectarnos a Oracle desde PHP necesitaremos instalar la extensión **oci8**, lo cual es más sencillo mediante un repositorio de extensiones como **PECL** y el instalador de paquetes **PEAR**.

```
$ yum install php-pear
$ pecl install oci8-2.0.12
```

Una vez tenemos la extensión instalada, para que PHP la utilice hay que especificarla en su fichero de configuración. Para ello editamos el fichero ubicado en `/etc/php.ini` y añadimos lo siguiente:

```
extension=oci8.so
```

Ya que estamos editando el fichero, establecemos también la zona horaria modificando o añadiendo la variable:

```
date.timezone = Europe/Madrid
```

Además, modificaremos la configuración de SELinux (módulo de seguridad del kernel de linux), en concreto el booleano que permite conectarse a servicios remotos, lo cual nos hará falta para conectarnos a la base de datos.

```
$ setsebool -P httpd_can_network_connect 1
```

Para evitar problemas de latencia al conectar con bases de datos externas, modificamos el fichero **/etc/resolv.conf** comentando o borrando los nameservers actuales (aunque posteriormente se añadan los que se necesiten realmente).

Ya tenemos instalado todo lo necesario y procedemos a crear el directorio donde se ubicará la aplicación y un usuario exclusivamente para gestionar los ficheros de la misma mediante SFTP “enjaulado”, lo que significa que sólo podrá tener acceso a la jerarquía de directorios que la contiene.

```
$ mkdir /var/www/docenciabd  
$ groupadd sftserver  
$ useradd -g sftserver -s /bin/false -d /var/www/docencia docenciabd  
$ passwd *****  
$ chown docencia:sftserver /var/www/docenciabd  
$ chmod 755 /var/www/docenciabd
```

Configuramos el demonio de SSH editando el fichero **/etc/ssh/sshd_config**. Comentamos la línea que empieza por “*Subsystem*” y añadimos lo siguiente al final del fichero:

```
Subsystem sftp internal-sftp  
Match group sftserver  
ChrootDirectory /var/www
```

```
ForceCommand internal-sftp
```

Reiniciamos el servicio de ssh:

```
$ systemctl restart sshd
```

Ya sólo nos queda configurar Apache para servir la aplicación, para lo que crearemos nuestro fichero *Virtualhost* que nos permitirá establecer la relación del nombre de dominio, en este caso *docenciabd.lcc.uma.es*, con el directorio de nuestra aplicación, que está ubicada en */var/www/docenciabd* y su directorio público es */var/www/docenciabd/publico*.

Creamos un fichero en el directorio **/etc/httpd/conf.d/**, con el nombre *vhosts.conf* por ejemplo y el siguiente contenido:

```
<VirtualHost *:80>
  ServerName docenciabd.lcc.uma.es
  Redirect / https://docenciabd.lcc.uma.es
</VirtualHost>

<VirtualHost *:443>
  DocumentRoot "/var/www/docenciabd/publico"
  ServerName docenciabd.lcc.uma.es
  SSLEngine on
  SSLCertificateFile /etc/ssl/certs/docenciabd.lcc.uma.es.crt
  SSLCertificateKeyFile /etc/ssl/certs/docenciabd.lcc.uma.es.key
  SSLCertificateChainFile /etc/ssl/certs/DigiCertCA.crt
    <Directory "/var/www/docenciabd">
      AllowOverride All
      Require all granted
    </Directory>
</VirtualHost>
```

Con esta configuración de Virtualhost forzamos a que todas las peticiones a la aplicación se realicen de forma segura, redirigiendo el tráfico HTTP a HTTPS para permitir solo este último.

Para poder cifrar la conexión con SSL serán necesarios un certificado firmado, la clave privada utilizada para generar la CSR (solicitud de firma del certificado) y el certificado de la autoridad de certificación que ha firmado el certificado.

La clave privada y el certificado se crean con el comando:

```
$ openssl req -new -newkey rsa:2048 -nodes -keyout  
docenciabd.lcc.uma.es.key -out docenciabd.lcc.uma.es.csr
```

El fichero CSR generado se envía a la autoridad certificadora y esta nos remite el certificado firmado junto con el certificado de la autoridad certificadora.

Apagamos la máquina y la exportamos desde VMware Workstation en formato OVF, que es el estándar más extendido para distribuir servicios virtualizados y permitirá la importación de la máquina en el hipervisor ESXi.